



RBPMN: the value of roles for business process modeling

Tarek Skouti¹ · Ronny Seiger² · Frank J. Furrer¹ · Susanne Strahringer¹

Received: 10 January 2023 / Revised: 22 May 2024 / Accepted: 26 July 2024
© The Author(s) 2024

Abstract

Business process modeling is essential for organizations to comprehend, analyze, and enhance their business operations. The business process model and notation (BPMN) is a standard widely adopted for illustrating business processes. However, it falls short when modeling roles, interactions, and responsibilities within complex modern processes that involve digital, human, and non-human entities, typically found in cyber-physical systems (CPS). In this paper, we introduce Role-based BPMN (RBPMN), a standard-compliant extension of BPMN 2.0 that distinctly depicts roles and their interactions within business processes. We underscore the value of RBPMN and a role-based context modeling approach through a modeling example in CPS that facilitates the representation of role-based variations in the process flow, namely a production process in a smart factory. Our findings suggest that RBPMN is a valuable BPMN extension that enhances the expressiveness, variability, and comprehensiveness of business process models, especially in complex and context-sensitive processes.

Keywords Business process modeling · Role modeling · Cyber-physical systems

1 Introduction

Business processes (BPs) are becoming increasingly complex, especially in our modern interconnected world where not only humans collaborate in BPs. Nowadays, we find highly complex human–human, human–machine, and machine–machine interactions in the same BP [1]. Such interactions are typical for cyber-physical systems (CPS), where many human and non-human process participants interact, each with their own process goals. Because modern CPS contain many physical and digital components—from single sensors and actuators to complex machines and appliances such as smart cars—it is beneficial to model their behavior and interactions in BPs, not as individual state

machines or sequence models for each component [2]. One example CPS that highlights this is a smart factory. We used the fischertechnik Training Factory 4.0 as a model smart factory, which we show in Fig. 1.

The individual production stations of the smart factory are interconnected, enabling the realization of complex business processes involving physical and digital performers. Exemplary processes in the model factory consider (1) the storage of workpieces in a high-bay warehouse with information about the workpieces (e.g., their color and ids) determined by different sensors and stored in a database; and (2) the processing of workpieces by an oven and milling machine with a robot being the central entity for transportation and various sensors to determine the status of the production stations [4].

Although these processes are machine–machine communication, they can be modeled with the de facto industry standard for business process modeling BPMN 2.0.

With recent research on the intersection of business process management (BPM) and the Internet of Things (IoT)/CPS [2], BPMN 2.0 has also been shown to help model simple processes in CPS [5]. However, modern business processes have become increasingly complex, revealing the limits of the BPMN 2.0 standard to express the necessary adaptability, flexibility, and context-dependence of the current business environment [6]. By taking BPMN 2.0 one step further with a standard-conformant extension that

Communicated by Selmin Nurcan and Rainer Schmidt.

✉ Tarek Skouti
tarek.skouti@tu-dresden.de

Ronny Seiger
ronny.seiger@unisg.ch

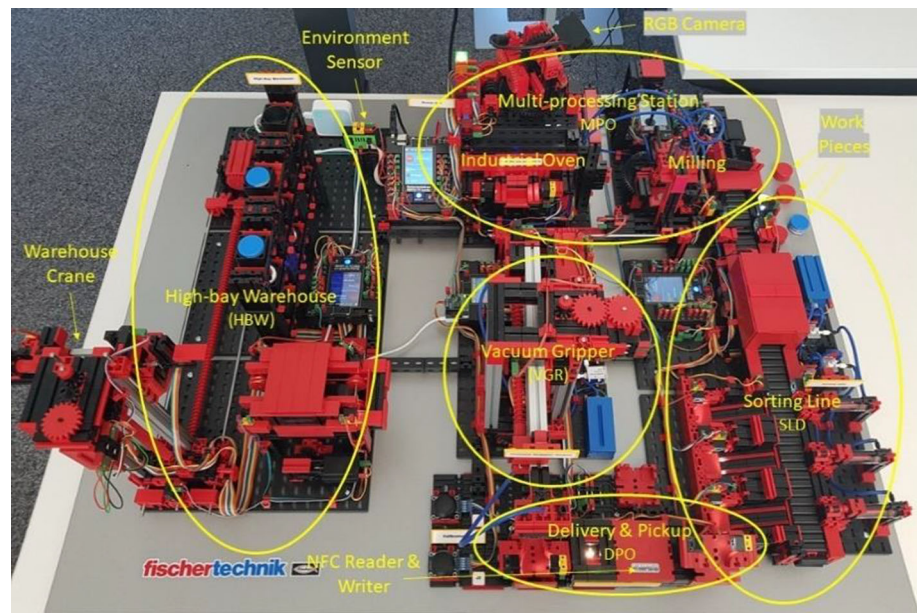
Frank J. Furrer
frank.j.furrer@bluewin.ch

Susanne Strahringer
susanne.strahringer@tu-dresden.de

¹ Technische Universität Dresden, Dresden, Germany

² University of St. Gallen, St. Gallen, Switzerland

Fig. 1 Fischertechnik smart factory model example [3]



explicitly represents context-sensitive and relational roles and their interactions in BPs (Role-based BPMN, RBPMN), we propose an approach that tackles many of the new process modeling challenges that arise with the increasing complexity of CPS and in other complex settings [2].

This work is an extension of the paper “RBPMN—The power of roles for business process modeling with BPMN” [7], published in the BPMDS 2022 Working Conference proceedings. This paper extends the scope of the original paper in the following ways: It provides more background information on roles and variability. The related work section includes modeling approaches for CPS and variability. The semantics of the RBPMN elements are stated explicitly. In addition to the original paper, we present a context-sensitive BP variability modeling approach. Finally, this paper presents a more exhaustive evaluation of the modeling language.

This paper is organized as follows. Section 2 presents the challenges modern BP modeling faces, the research objectives we derived from them, and our methodology and how we aimed to fulfill the objectives. Section 3 presents the role concept, process variability, and CPS background. Section 4 presents related work in the problem area. Section 5 details our role-based BPMN extension RBPMN, its development, syntax, and semantics. Section 6 introduces the new context-sensitive approach for process variability, and its application in RBPMN is demonstrated with an example in Sect. 7. Section 8 shows the evaluations we conducted. Section 9 discusses our findings and limitations before Sect. 10 concludes our paper and presents an outlook.

2 Objectives and research methodology

2.1 Motivating example

We investigated the inbound delivery process from our model smart factory (cf. Fig. 1) in more detail and modeled it with BPMN 2.0.

The process starts when the light barrier is interrupted by a workpiece placed into the delivery and pickup output (DPO) station. This triggers the system to get the current environment status and to register a new workpiece in the database. The vacuum gripper crane (VGR) then moves to pick up the workpiece from the delivery and pickup (DPO) station. At this point, there are three options to determine the color of the workpiece, each through a different sensor—a color sensor, a camera, or a brightness sensor—given preference in this order if the previous one is not available. Thereby, the color sensor is preferred due to its fit for purpose, processing the data from the camera is more expensive, and the brightness sensor is less accurate. (1) The first option requests a color sensor in the delivery and pickup station. The VGR will move the workpiece from the light barrier to the color sensor and place the workpiece on the color sensor, where the activity read color is performed and the result is written to the database. The VGR then picks up the workpiece again. (2) The second option is using the RGB camera. The VGR moves the workpiece to the camera so the system can take a picture, send the picture, and then analyze the picture to determine the color. The color is then stored for the workpiece in the database.

(3) The third option makes use of the brightness sensor. The VGR moves the workpiece to the brightness sensor,

which sends value changes in the environment status. The system measures the difference in brightness based on the reflection of the workpiece and calculates the color on the brightness changes. This information is then written into the database.

After the color is determined, the workpiece is moved to the NFC reader/writer if it is available as determined through the request NFC sensor activity. The VGR places the workpiece on the NFC sensor, which reads the information already stored on the workpiece's NFC tag. The system writes new information including the time of delivery, environment status, color, and product id within the system to the workpiece. It is then again picked up by the VGR, which moves the workpiece to the next station. The goal of the inbound delivery process is to register an incoming workpiece in the system, determine the state of the workpiece, pick it up, and move it to the next station.

Now, each of the sensors used for color detection can fail. The color cannot be determined if all sensors fail or the system fails to communicate with them. Without knowing the color or additional information, the process should not move forward for individual production but instead, switch to a different batch processing process rather than customization. For example, without knowing the color of the workpiece, a custom request for a red item cannot be fulfilled. However, a batch process (mass production) that does not require individual information is still possible, and a sold item would generate revenue for the business.

A human worker in this process could also identify the color and manually edit the database. If we make the assumption that the workpiece cannot be carried or moved by a human, the added complexity is manageable. If we assume that a human could carry the workpiece to the next station, the complexity of different potential paths would explode. As we wrote above, we found the process model of just a few variants of a rather straightforward process already to be complex and large. The additional human process participant does not add value in the example.

In our example, we assume that the NFC reader/writer would only be relevant for custom productions because customers would want updates on their customized products. If the color cannot be determined, a customized product, which demands a specific color, is impossible. Henceforth, the activities involving the NFC reader/writer could be skipped.

The inbound delivery process is relatively straightforward, as the goal and the steps to achieve it are known at design time: A workpiece gets delivered, is registered, its state is determined and written into a database and onto the NFC tag of the workpiece and then moved to the next station. Yet, the resulting process model is complex and hardly readable because there are many inputs and outputs from the numerous sensors and actuators to consider. A well-readable

process model should have 5–7 activities according to Silver's Method and Style Guide [8]. If the model is executable, the number of activities increases manifold. Our model in Fig. 2 has 29 tasks and only covers a few process variants. Although we consider the process to be relatively straightforward, the complexity is already very high. Furthermore, expressing each sensor as its own lane increases the size of the BPMN 2.0 model, making it less readable.

Modern BPs, including those in CPS, are highly complex [9]. Their process flow must adapt to sudden changes during execution, requiring the BP model to reflect the required implementation accurately [8, 9]. However, the CPS domain is not the only one that presents novel challenges to BP modeling.

2.2 BP modeling challenges

Modern BPs are complex collaborations of humans and machines. Modeling them has brought forth many challenges. We investigated recent research manifests, practitioner surveys, and scientific literature to derive a non-exhaustive list of challenges that pertain to our motivating example and to the modeling of complex business processes in domains such as IoT and CPS [2, 5, 10, 11]. We used these sources as a basis to derive a more generalized list of five business process modeling challenges C1–C5 that emerge with the modeling of BPs of increasing complexity in knowledge and data-intensive domains such as CPS, and where introducing the concept of Roles might be beneficial to address these challenges. The identified five challenges are of high relevance when modeling complex business processes that involve multiple stakeholders and performers, not only in the domains of CPS and IoT. Furthermore, we categorize these challenges if their solutions will require higher expressiveness, better comprehensiveness, or more process variability from a BP model.

2.2.1 C1 Expressing different process performer types (expressiveness)

Today, machines often include AI-enhanced devices (Challenge Smart Devices in [12]). We use the term AI-enhanced instead of smart devices to cover devices that used to be “dumb” (no connectivity) but were upgraded and can now be controlled through a system or AI. They can participate as active process performers who execute tasks and change the state of a thing, and they can participate as passive process performers, like sensors, that determine the state of a thing [12]. Our example (Fig. 2) does not express sensors, actuators, or the system differently. Furthermore, the data object is insufficient to express the different types of data within a modern process (Challenge 5 in [12]). A modeling approach should be able to express new process performer types that

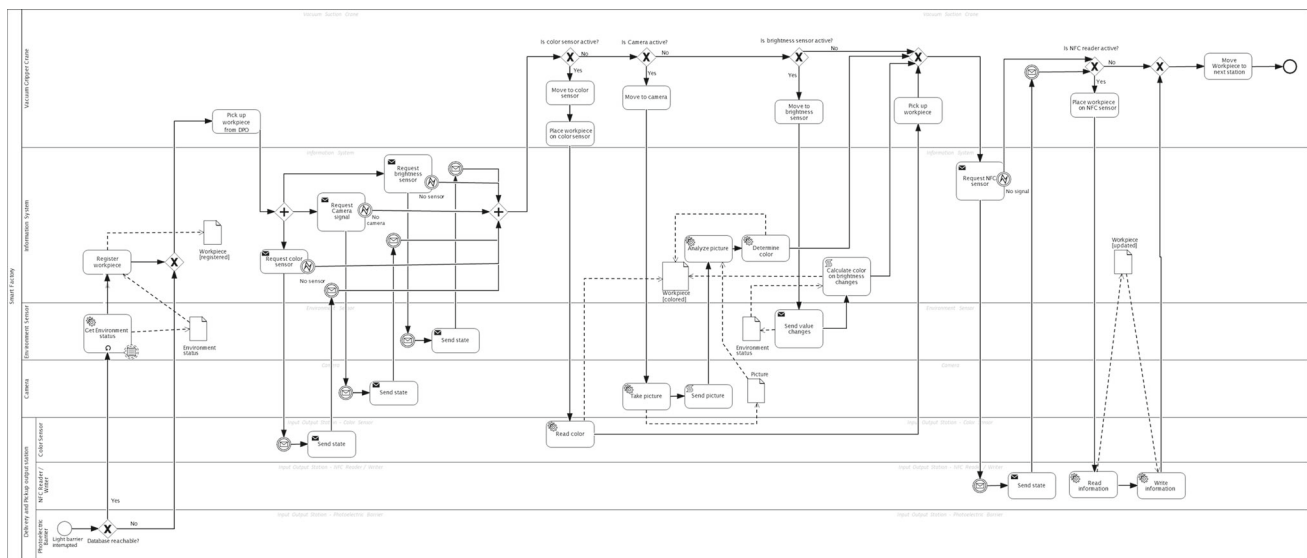


Fig. 2 BPMN 2.0 model of the inbound delivery process (zoom in the electronic version or find it at publisher's website)

might emerge with new technology. A recent systematic mapping study shows that many researches elect to introduce new constructs to modeling standards to express IoT devices, as the existing modeling elements are insufficient [5]. These different challenges all relate to expressiveness and expressing different process participants.

2.2.2 C2 Unifying stakeholder perspectives (comprehensiveness)

To effectively and correctly implement a system it is important to consider the perspectives of all involved stakeholders. BP models describe a system from a behavioral perspective, but there are other perspectives, such as the system structure, that must be taken into account. Suppose stakeholders with different perspectives understand each other's models and concepts. In that case, they can form a unified understanding of the envisioned system and processes (Described as Challenge 6 in [12], Perspective Agility in [13], and expansive BPM and fixed granularity levels for process analysis in [14]). Unifying the stakeholders' perspectives requires an increased ontological overlap between them [13]. An increased ontological overlap will also help to ensure that the BP is accurately implemented from the BP model [12], which is especially important to keep alignment with the structural aspects of the system (Described as Challenges on modeling software systems from different viewpoints in [15]). These challenges all relate to increasing the comprehensiveness of the process model across multiple stakeholders.

2.2.3 C3 Coping with BP model complexity (comprehensiveness)

Many different languages can be used to model BPs, with varying levels of complexity. For example, the de facto industry standard BPMN 2.0 is a complex BP modeling language with 90 objects and 143 properties [16]. For comparison, the UML activity diagram has nine objects and six properties [17]. However, using a complex modeling language does not necessarily result in a complex BP model, as BPMN scores best for the readability of its models [18]. Of course, a BPMN model usually contains just a fraction of all BPMN 2.0 modeling elements. However, those elements that are required for BP simulation and execution are often missing in stylized models, which are used for BP analysis [19]. For example, our motivating example (Fig. 2) is not a stylized model and is already complex, even though the process is simple. One reason is the “the lack of objectivity in process descriptions” [14]. Keeping the complexity of a process model low will increase the comprehensiveness of the process.

2.2.4 C4 Modeling BP variability (variability)

Modern BPs are complex, ever-changing, and evolving [20]. Capturing this evolution and variability of a BP during run-time requires ad-hoc changes to the BP structure, as the actual BP structure might be unclear at design time (Described in Perspective Agility in [13] and in Challenge 8 in [2]). Current BP modeling standards like the BPMN 2.0 are suitable for BPs with low variability but insufficient for BPs with high variability [9]. The presented challenges relating to variability neither fit the category comprehensiveness nor expressiveness.

2.2.5 C5 Modeling context-sensitive elements (expressiveness and variability)

BPs are not context-independent. In many cases, changes to the context should trigger a response from the process flow. If a sensor fails, less context is perceived. Even though the objective context has not changed, less information is available to achieve the process goal. Therefore, the process flow should adapt [21, 22]. For example, if the color sensor of our model smart factory fails, the process should continue and not terminate. Instead, the process sequence now uses the camera to determine the color of the workpiece. The analyzed literature refers to this challenge as Paradox 7 [2], Automated Process Adaptation [13], and Context-aware Execution [23]. While this challenge relates strongly to modeling variability, it also relates to the expressiveness of context-dependency in a model.

In summary, current BP modeling languages are, on the one hand, too complex and, on the other hand, still not expressive enough for dynamic processes that use AI-enhanced or complex data objects. The BP models need to express context-sensitivity and are hard to unify with models of other modeling perspectives. If we look again at our motivating example, expressing only some variability already results in a large and complex model (cf. Fig. 2).

2.3 Research methodology

Simply reducing the complexity of our motivating example would not bring merit to the scientific community and practitioners. Therefore, when aiming to solve the problems of context-sensitive adaptability, model complexity, and missing concepts to address the variety of performers, we must inevitably address the challenges C1–C5 through research. Based on these challenges, we formulate our research questions as follows:

RQ1: How can the expressiveness, variability, and comprehensiveness of business process models be improved?

RQ2: How can context-sensitive variability in business process models be expressed?

From these research questions, we derive the following research objectives:

- O1: Increase the expressiveness of BP models to model a variety of process participants (C1)

Addressing challenge C1, our objective (O1) is to accurately enhance the expressiveness of BP models to represent diverse process participants, including AI-enhanced devices.

- O2: Increase the overlap of the structural and the behavioral modeling perspectives (C2)

Given challenge C2, our objective (O2) is to enrich BP models by merging structural and behavioral perspectives, facilitating a unified system understanding. We chose the structural perspective as it is an essential and well-established system modeling perspective.

- O3: Increase comprehensiveness of BP models through modeling constructs that are well understood in the real world (C3)

The complexity of BP models presents another challenge (C3), with different modeling languages offering varying levels of complexity. Therefore, our third objective (O3) is to increase the comprehensiveness of BP models by incorporating modeling constructs that are well understood in the real world, thereby making the models more accessible and easier to understand.

- O4: Enhance variability modeling for context-sensitive processes (C4, C5)

The last two challenges (C4 and C5) relate to the need for modeling BP variability and context-sensitive elements. Modern business processes are complex, ever-changing, and often context-dependent, which requires models that can capture this variability and adapt to changes in the context. Accordingly, our fourth objective (O4) is to enhance variability modeling for context-sensitive processes, allowing for the representation of dynamic and evolving business processes.

We position our research in design science with a modeling language and modeling style as the resulting artifacts [24]. In particular, we follow the design science research methodology suggested by Peffers et al. [25] see (Fig. 3). It consists of 6 phases, starting with the motivation phase, where a problem is identified. Our motivational example (Sect. 2.1) and the challenges (Sect. 2.2) represent this phase. The second phase defines the objectives of a solution to the abovementioned challenges C1–C5. We are in particular looking at objectives that address overlap between modeling perspectives, process variability, and CPS (O1–O4). These terms are explained in Sect. 3. The third phase is the design and development of a solution. Design builds on top of existing approaches, which is why we present related approaches that inspired us in the related work (Sect. 4). We extend the de facto standard for BP modeling, the BPMN 2.0 [16], to accommodate the abovementioned challenges. The resulting artifact is our Role-based BPMN (RBPMN). The fourth phase is a demonstration of the artifact, for which we use our motivational example. The fifth phase is evaluation, for which we provide five types of evaluation (Sect. 8). The workflow

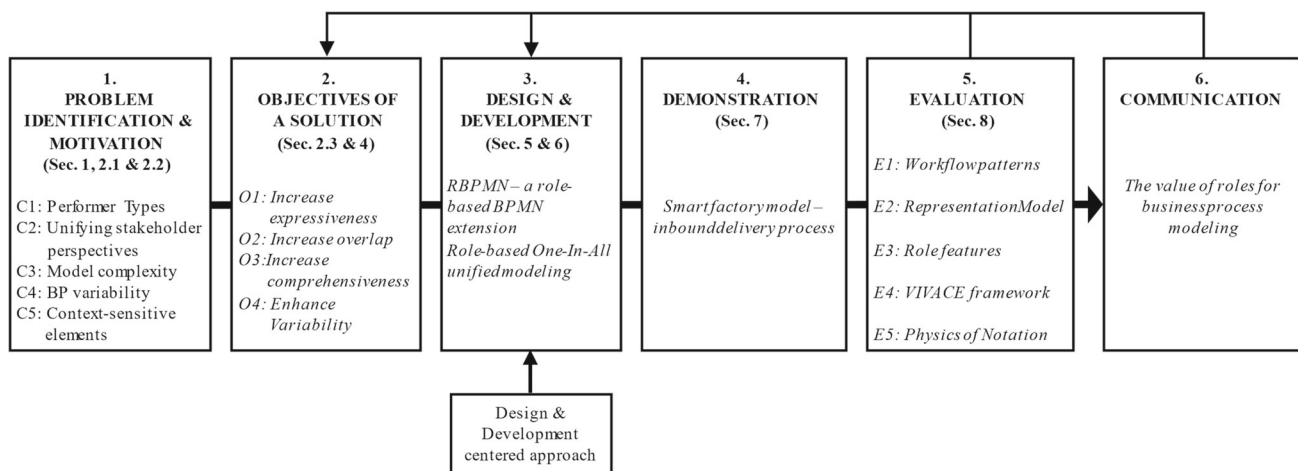


Fig. 3 The Design Science Process of this work (adapted from [16])

patterns, the representation model, and role features evaluate the expressiveness of RBPMN. The VIVACE framework is used to evaluate variability in RBPMN. We make a theoretical argument based on Moody’s “Physics of Notation” for the comprehensiveness of RBPMN. Finally, the last phase of the design science research methodology is the communication of the artifact, which is this publication.

3 Background

This section introduces the background to a solution that satisfies our objectives. The most relevant concepts here are Roles, Process Variability, and CPS—as our smart factory example is one. We investigate the application of Roles as the basis for a model unification approach because roles have a relational, behavioral, and context-sensitive nature [26], are applied in a variety of modeling perspectives [27], and have been applied to increase the adaptability of structural models [28] and CPS [29]. A model unification combines different models into a coherent framework to create a more comprehensive representation for multiple stakeholders. Thus, they may also prove to be beneficial for modeling BPs to address the objectives mentioned in Sect. 2.

3.1 Roles

Bachman and Daya defined a role as a “behavior pattern which may be assumed by entities of different kinds [30]”. This already indicates that it is not just humans that play roles. While roles were originally introduced for data models, they have since found their way into all aspects of software development because “The concept of role is such a fundamental one that it is hard to avoid it when talking about the world.

[31]”. We can find roles in behavior models [7, 32–35], structural models [28, 36, 37], data models [38], security models [39], and more [37]. As we explain the application of roles in process models in the related work section, we use this section to discuss their application in other modeling perspectives.

3.1.1 Roles in data models

Bachman and Daya’s definition of roles decoupled behavior from the entity’s structure. An entity had to play a role to be a member of a relationship and drop the role to leave the relationship. Therefore, relationships were defined between roles, not entities [30]. The Entity-Relationship (ER) model and the Object-Role Modeling (ORM) language are other role data models to name here. ER and ORM define roles as descriptions of relationships [38]. However, the definitions in ER and ORM do not make good use of the role concept or its power of expressiveness.

3.1.2 Roles in security models

A famous security approach is role-based access control (RBAC). In RBAC, roles are understood as: “[...] a set of transactions that a user or set of users can perform within the context of an organization [39]”.

3.1.3 Roles in structural models

In structural object-oriented models, roles introduce behavioral and temporal awareness [28, 38]. For both the concept class is too static to represent them. Bera et al. [38] define that: “a role can be viewed as a functional schema whose functions represent mutual properties arising out of an interaction among instances of one or more classes.” Building

upon advances in role research, Schön [36] defines a role as “A contextual modeling construct with state and behavior that is fulfilled by an object or its roles to represent it in the user’s context and extend or change its corresponding specifications and interactions.” Roles were used for adaptation purposes and to provide some dynamic perspective in structural modeling [28]. One language we want to name here specifically is the business role-object specification (BROS), which incorporates more recent advances in the role research stream.

3.1.4 Role characteristics

Even though all perspectives can use the role concept, this did not lead to a consistent understanding of roles over multiple perspectives. Role research suffered from discontinuity, leading to investigations of the existing role definitions and whether they have a coherent core [31, 37]. They do, and the coherent core is anti-rigidity and context-dependence [26]. The following example explains this coherent core. An object can only play the role of Buyer if another object plays the role of Seller. Both roles only exist in a shopping context. When an object leaves the context or drops the role, it can no longer buy or sell. The objects fulfilling these roles would still exist, but not in these roles, making the performer rigid and the role anti-rigid [37].

Besides these role attributes, Steimann derived 15 role features [31], which Kühn later extended to 27 [26], that characterize role-based modeling languages. They found that rarely all role features are considered, as there is no single definition of roles [26, 31]. Therefore, a researcher developing a role-based modeling language should clarify their definition of the term role [38].

3.1.5 Role definition

Throughout this paper, we define and understand a role as: “A contextual modeling construct with state, behavior, and sensory power that is fulfilled by an object or its roles to represent it in its context and extend or change its corresponding specifications and interactions.”

One aid for defining role modeling languages is the compartment role object model (CROM), a feature-complete metamodel for role-based modeling languages [26]. The development of a role modeling language for any modeling perspective with CROM as the basis allows a relation to other role modeling languages.

3.2 Business process variability

The second topic we want to provide more information on is process variability. Process variability is a pervasive issue in

modern BPs and one of the reasons for their increasing complexity. BPs often vary between different countries (due to regulations), different domains (due to different standards), different customers (premium or not), and so on. Every new context can result in a new BP variant. Context-sensitive BPs especially possess many variants to achieve their process goal, which is why we explain this topic in more detail here. The core BP is typically the same for the different BP variants. Some tasks or flows are slightly different between the variants [40]. Modeling process variability is the endeavor to capture these variants in either a set of related models or in one complex model [41]. Managing these variants is not trivial, so different approaches to dealing with variability were brought forth.

One approach is configurable process models that define variation points within a model. Variation points define where variability can occur, and in a model, they can be specified through annotations to tasks [42]. A different approach to model variation points is configurable nodes. A configurable node is associated with multiple configuration alternatives. A configurable node adds variability to a model because the specific process variant is derived once all configurable nodes are set [43].

When working with reference models from which organization-specific models are derived, model projection is projecting only the information of a reference model relevant to a specific scenario. Configuration parameters can exclude elements of a model or even element types from the reference model [44]. However, an additional adaptation step is necessary to achieve an organization-specific model, as the configuration is limited by the reference model [42].

Hiding and Blocking are the next approaches we want to address. Blocking removes edges from a process, which disables some activities as they and their subsequent tasks can no longer be reached. Hiding keeps the path in the process but removes the activity so that subsequent tasks can still be executed [45].

Other variability mechanisms are fragment customization, where process fragments are added, deleted, or modified; and activity specialization, where activities of a model are replaced through specialized versions [6, 40].

3.3 Cyber-physical systems

A CPS consists of a cyber part, typically a software system, that controls and operates the physical part of the system. Sensors of the physical system feed the cyber system, which uses the sensor values to determine appropriate action and operates the physical system through actuators [46] (Fig. 4).

A related term is “Internet-of-Things” (IoT). Almost all IoT processes are either a Process-of-Things or a Things-aware process [10]. Our motivational example of a CPS follows the notion of Things-aware processes. Things-aware

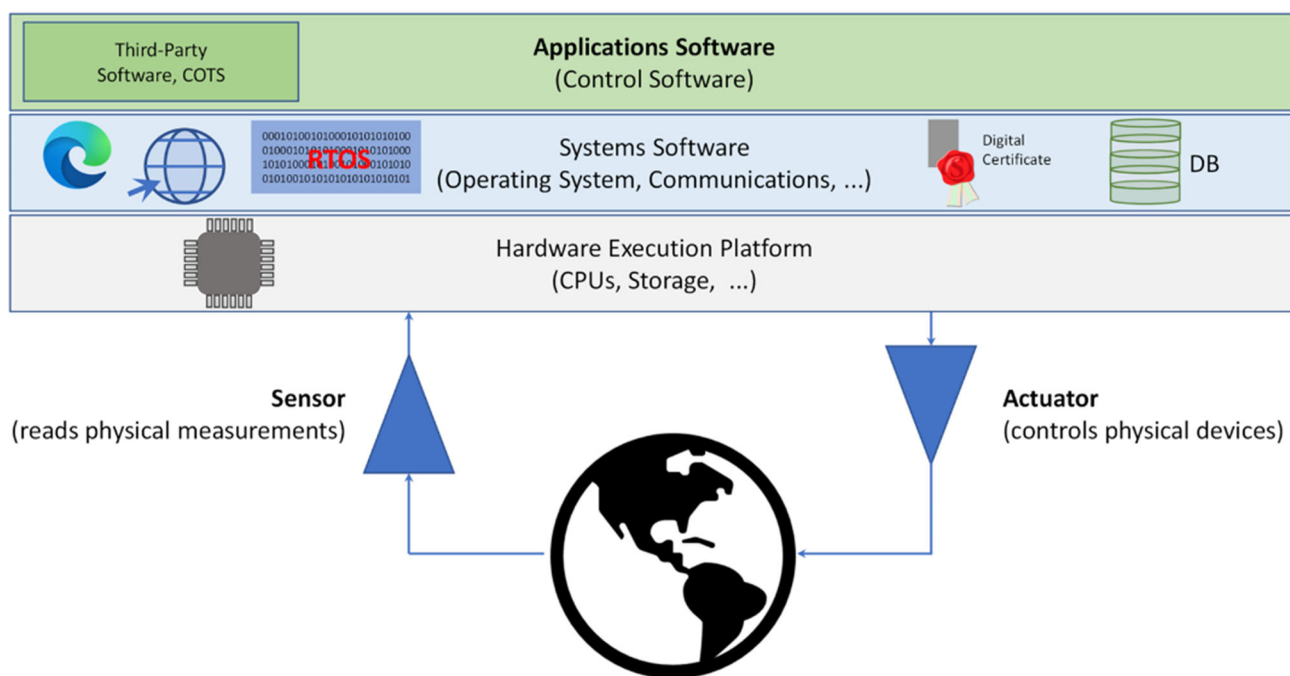


Fig. 4 Cyber-physical system (adapted from [2] p. 4)

processes integrate (AI-enhanced) devices into the system instead of focusing on the inner workings of a device [10]. The IoT-A project has defined the general requirements for IoT processes, which include privacy, interoperability, security, and many more (over 180) [47]. However, IoT can also provide benefits for BPs and vice versa if concerning challenges are resolved. These challenges were defined in the BPM-IoT manifesto [2]. BP models can help manage the complexity of IoT processes, coordinate many participants, support them in dealing with new situations, and adapt existing models to new contexts. The large amount of sensor data can support conformance checking [2]. However, within a CPS, AI-enhanced devices do not necessarily communicate just with one another but also with humans [48].

A CPS has diverse components that participate in a BP [49]. Various approaches exist to model this diverse range [5, 10]. However, they have no individual modeling constructs in the modeling standard BPMN 2.0. Furthermore, BPMN 2.0 cannot differentiate between cyber and manual tasks and cannot sufficiently express process participants who do not perform physical tasks [11]. In addition, a BP model should express error handling and how resources are handled at run-time [11].

4 Related work

4.1 Role-based process modeling

Roles have been part of business process modeling languages in different forms and with different definitions for many years.

Some of the earlier approaches that considered roles as sets of sequentially ordered actions and or interactions are the role activity diagram, the role interaction nets, and the object-oriented role analysis method [33]. However, as processes became more complex and knowledge workers started to rise, a sequential activity order was not always given. Knowledge processes require an evolved definition of roles because activities are performed independently of their sequence. Therefore, [33] defined roles as the process knowledge of a process participant. An open question that remained was who is responsible for a task, is the performer playing a role, or is the role that executes the task? The flexibility of a BP is enhanced if tasks are not assigned to performers but to roles. It is the responsibility of whoever plays a role in that process instance, not of a specific performer [32]. Other process modeling languages that focus on the behavioral and functional perspective, like UML activity diagrams and BPMN, also contain some role characteristics [50].

The de facto standard for business process modeling, the BPMN, specifies two types of roles: Resource Role and Process Role [16]. The resource role can be responsible for the process or an activity. A specialization of it is the human

role, to clearly define that the performer type for this role must be human. Other potential performers of resource roles are a group (e.g., sales), organizational role (e.g., manager), or an organization (e.g., supplier) [16]. What is missing from this definition of potential performers are non-human AI-enhanced objects. The process role describes an organizational (e.g., manager) or business role (e.g., buyer). The process role is expressed through the notation element Lane. A resource role can be associated with a process role, but the resource role is not explicit in a process model [16]. This view of roles does not incorporate all the advances role research has made [38]. Luckily, BPMN 2.0 can be extended, and this was done so to integrate a resource perspective [51, 52], to bring context more prominently into the spotlight [53, 54], and to incorporate roles [7, 35, 48, 55].

The RALph approach extended BPMN by adding multiple new connectors that express whether a resource (performer) should remain in a role for subsequent activities [51]. Another approach to incorporate the resource perspective is to add a resource diagram that can be referenced in the process model, which the BPMN4CP extension does [52]. The actual process flow always depends on the context, which is why it needs to be present in some form in the process model.

C-BPMN adds a context model to BPMN process models to define which variant to use, depending on the current context. Changes to the context lead to a switch to another BPMN model in C-BPMN [53]. The BPMN-CARX extension introduced a global process, task, and Augmented Reality context to BPMN models for IoT and Augmented Reality, where each context specifies rules. The global context contains organizational rules that must be executed before a task. The process context contains process-specific rules that can override the global rules. The task context contains task-specific rules that override the other rules and contains additional information necessary for the task execution, like IoT sensor data or potential hazards. The Augmented Reality context contains specific information for displaying elements correctly [54].

Even though the BPMN 2.0 standard uses roles, their application does not use what roles can provide. Therefore, several BPMN 2.0 extensions add roles to some extent. BPMN4CPS extends the standard for CPS by adding device roles (sensor role and actuator role). Because modeling each device as a separate lane would result in an abundance of lanes, BPMN4CPS proposes using color to define different roles, thereby ensuring that the model remains readable [48]. The BPMN 2.0 extension of [40] presents a different approach to using roles. They use roles to define task-authorization constraints. Tasks are assigned to roles, and users play these roles in a workflow. It could be predefined that each user must remain in their role during workflow execution. This ensures that conflicting tasks cannot be performed by the same person, like writing and reviewing one

paper. The extension's authors use text annotations to specify the authorization constraints [55].

Recently R-BPMN, a role-based BPMN that applies roles for abstract modeling of process patterns, was presented. R-BPMN describes patterns in terms of roles for reuse in different processes. Every model element could potentially play a role and be used to define an abstract pattern [35].

We can see that as BPs evolved, so did the role understanding used to model them. Roles are already used to model CPS (BPMN4CPS [48]), to unify behavior with different modeling perspectives (RALph [51] and BPMN4CP [52]), and to enhance context-sensitivity (C-BPMN [32]). Furthermore, BPMN 2.0 was already extended with roles but for a different research goal (R-BPMN [35]). Nevertheless, we saw a mismatch between the advances role research made and the application of roles in process modeling. Roles were used to address individual challenges but not as a solution to solve multiple challenges in different areas.

Therefore, this paper proposes a role-based BPMN based on the contextual and relational role definition propagated in the compartment role-object model (CROM) [26]. This ensures, first and foremost, that the resulting role-based modeling language can be integrated with other role-based modeling languages that adhere to CROM. Furthermore, our challenges and research objectives span multiple domains, meaning a satisfying solution is not just domain-specific. We expect the benefits that roles have proven for structural modeling [36] to be transferable to process models. Previous approaches were satisfactory toward their respective research objectives and challenges. However, satisfying our research objectives would require the combination of multiple and then, in some part, contradicting extensions, which would increase the complexity manifold. With our approach we aim to satisfy challenges of multiple domains and to combine expressiveness with process variability without a significant increase of complexity.

4.2 BPMN extensions for process variability

We use this section to describe BPMN extensions that focus on process variability. A recent literature review compared BPMN-based variability approaches that deal with business process families [40]. We examined those approaches that extend the BPMN standard.

BPMN* uses feature models to define the node configuration of BPMN model elements. A model element can be a task, event, or gateway. The feature model defines the context for the different variants. The intended use is for a set of models [56]. The authors of BPMN* do not explain how they extended the BPMN metamodel to remain in the standard.

BPMNnext provides variability by defining activities as variation points. It provides a set of models derived from the core model's variation points. It can also be extended

to enable variability through gateways as variation points. The authors of BPMNNext highlighted how they extended the BPMN metamodel [57].

vBPMN extends BPMN to incorporate the rule language R2ML for variability. The rules are applied to derive process variants, resulting in a set of related BP models. The variability is expressed through process fragments with only one entry and one exit. The authors of vBPMN explain how they extend the BPMN metamodel [58].

The BPFM approach of [59] provides multiple variants and variation points within one model. Tasks are extended with an additional symbol that expresses if a task is optional. However, there is no context definition.

C-BPMN defines configurable process fragments for variability. The process fragments contain context conditions under which they are excluded, included, or optional. The authors of C-BPMN did not describe their extension of the BPMN metamodel but provided formal definitions [60].

BPMNt extends BPMN for tailoring software development processes. There are no representations of context, as tailoring defines just one variant. BPMNt does not provide new notation elements and, therefore, remains in the BPMN standard by adhering to the extension mechanism [61].

Overall, there are quite a few BPMN 2.0 extensions that enhance process variability and some that aim to incorporate roles and context. This shows that standard BPMN 2.0 is insufficient to cope with modern challenges. The existing approaches address either context-sensitivity, variability, or perspective unification well. However, they do not address all areas simultaneously. We aim to develop a BP modeling language using a modern understanding of roles (Sect. 3.1) to enhance process variability and context-sensitivity and unify the behavioral and structural perspectives.

4.3 Business process modeling in IoT and CPS

Using notations from the BPM domain to model processes in IoT and CPS has received increasing attention in recent years. Literature studies on modeling IoT-related aspects in BPM can be found in [5, 10, 23, 62, 63].

Many of the relevant approaches propose to introduce new types of activities [62, 64, 65], events [65, 66], and process resources [67–69] to represent IoT and CPS-related aspects covering sensing and actuation in business processes. The new IoT devices usually have different capabilities [70, 71] and contribute to executing one or more process activities, either alone or in collaboration with others [67, 71–73].

Moreover, related research discusses and addresses the abstraction gaps that come from different granularities of actuations and sensing in IoT [14, 74] that need to be represented in activities, events, and decisions in a business process model [11, 71, 75, 76]. Data are often produced by one or more IoT devices (sensors) in combination at a

much lower granularity than it can be used to make informed decisions on the level of a business process. To not increase the complexity of process models by, for example, representing each sensor as an individual lane in a BPMN-based process model, several works discuss the use of abstraction mechanisms such as complex event processing [75, 77] and ontology-based inference [71] to combine and abstract low-level sensor data.

All of the aforementioned approaches specifically target the modeling of novel IoT and CPS-related aspects that emerge with the extension of BPM into the physical world. They address specific challenges and objectives associated with the IoT and CPS domains, but do not cover the more generalized challenges and objectives that we have identified related to the increasing complexity and variability of business process models in Sects. 2.2 and 2.3. In our work, business processes in CPS (e.g., a smart factory) merely serve as illustrating examples, but complex business processes that may benefit from a role-based modeling approach can also be found in many other domains (e.g., finance, e-commerce, or healthcare).

5 RBPMN: a role-based BPMN

5.1 Developing RBPMN

The BPMN 2.0 standard and the abovementioned extensions do not address all our objectives with one modeling language or language extension. For the first objective, the presented BPMN extensions in the related work Sects. 4.2 and 4.3 were for specific domains or specific devices and did not provide a general solution for any new process participant. The related work does not provide a unified approach that effectively merges structural and behavioral perspectives in a way that enhances the overall system understanding, which is necessary for achieving O2. The related work in Sects. 4.2 and 4.3 does not sufficiently address the need for modeling constructs that are easily understood in the real world, which is essential for achieving O3. They are useful for domain experts but not for novices. Furthermore, many extensions introduce more complexity into the process model. Finally, the related work in Sect. 4 does not provide a holistic approach that enhances variability modeling for context-sensitive processes in a way that addresses multiple domains and reduces complexity, which is necessary for achieving O4. There is no approach that satisfies all objectives, each one individually is solved by at least one. We aim to close this gap so that instead of requiring modelers to use different languages for each objective, which just adds to the complexity of managing models, we see benefits in having one language that fulfills all objectives. Therefore, we were faced with the choice of either developing a new BP modeling language

entirely or extending an existing one. Both approaches have their advantages and disadvantages.

Developing a new language without limiting restrictions allows for better targeting of the mentioned objectives by modeling context-sensitive models with high variability that express different stakeholder perspectives. However, this does not take into account opposing goals or objectives. A new language could also address issues of existing languages, like ontological clarity or overload of concepts [78]. The downside is that developing a new language adds to the existing plethora of languages. In addition, language adoption in practice without commercial tools and toolchains is unrealistic [79].

Extending an existing language that is already widely used enables practitioners to continue using their tools. Furthermore, existing semantics can be used. While there are no experts in the extended language, there are experts of the standard who can quickly become experts in the extension. The downside of extending is the restrictions to be considered in order to remain within the standard [79]. This also means that not all requirements might be satisfied.

We chose to extend an existing language, to be precise, the de facto standard for business process modeling BPMN 2.0. Our role-based BPMN extension (which we termed RBPMN) was developed to provide multiple benefits. To increase the expressiveness, we used Roles as the key modeling construct. In particular, we used the modern role understanding propagated through CROM, which can already be found in different modeling perspectives, and introduced it to BP modeling [26]. Because our role understanding (see Sect. 3.1) sees roles as behavioral, relational, and context-sensitive constructs, a broader introduction of them to BPMN 2.0 will bring context more into the spotlight.

5.2 Extending BPMN

The BPMN 2.0 specification describes an extension mechanism by addition to develop standard conform extensions. The problem with the extension mechanism is that it is not well described. It states that the semantics of any BPMN 2.0 element must not change and that a resulting extended BPMN 2.0 model should retain the look and feel of a standard BPMN 2.0. This led to extensions that are not standard conform [80, 81]. The BPMN + X approach of Stroppi et al. guides the development of standard conform domain-specific BPMN 2.0 extensions [82]. However, our approach is not domain-specific but of an overarching nature, which is why we chose the evolved BPMN + X approach of Braun and Esswein [52]. They extended BPMN 2.0 to integrate a resource perspective. As one development goal was unification with other perspectives, it was deemed the more promising approach. Figure 5 provides an overview of the method.

We explain its application in the following subsections. The approach consists of 4 phases. It starts with developing a conceptual model of the extension, which we show in Fig. 6. The extension contains elements of CROM [37] and the role-based structural modeling language BROS [28] (Red), BPMN [16] (green), and new elements (orange) to address the challenges in Sect. 2.2.

5.3 Abstract syntax

The transition from the conceptual domain extension model (CDME) to the BPMN + Extension model requires a description of the RBPMN in terms of the BPMN 2.0 extension mechanism (see Fig. 7). The 15 transformation rules to derive a BPMN 2.0 extension from a conceptual model (cf. Table 1) are taken from [82].

These rules present mapping guidelines to ensure the BPMN extension conforms to the mechanism. For example, suppose a new data type of an existing BPMN concept is modeled in the CDME model. In that case, this new data type is an *ExtensionAttributeDefinition* in the BPMN + X model. If an entirely new concept is modeled in the CDME, it is transformed into an *ExtensionElement Property/ExtensionAttributeDefinition*. Furthermore, suppose the new concepts are generalizations or specializations of existing BPMN concepts in the CDME. In that case, they are transformed into *ExtensionRelationship* in the BPMN + X model.

Following these rules results in a BPMN + X model, the abstract syntax of RBPMN (Fig. 6). We keep the standard BPMN 2.0 elements green and the extension elements orange. We show only those BPMN 2.0 elements that are related to RBPMN elements. The element *Pool* already fits the compartment definition as an “objectified collaboration with a limited number of participating roles and a fixed scope [37]”. Therefore, we did not transform it. The passive role is an extension concept relating to the BPMN concept artifact. From Table 1, we can deduce that an *Extension Concept* of the CDME model is transformed into an *ExtensionElement* in the BPMN + X model. The *Role* as an *Extension Concept* and super class of active and passive roles is transformed into an *Extension Definition*.

The third phase addresses the development of the concrete syntax, which we explain in the next section. The subsequent section will then present the semantics of the RBPMN elements.

5.4 Rationale and semantics

We use this section to explain the semantics of the new RBPMN modeling elements, which we show in Table 2. The new elements address the research objectives O1, O2, and O3. The new attribute of *BaseElement* *ContextLevel* is needed to

Fig. 5 Adapted overview of the extension method of Braun and Esswein [24]

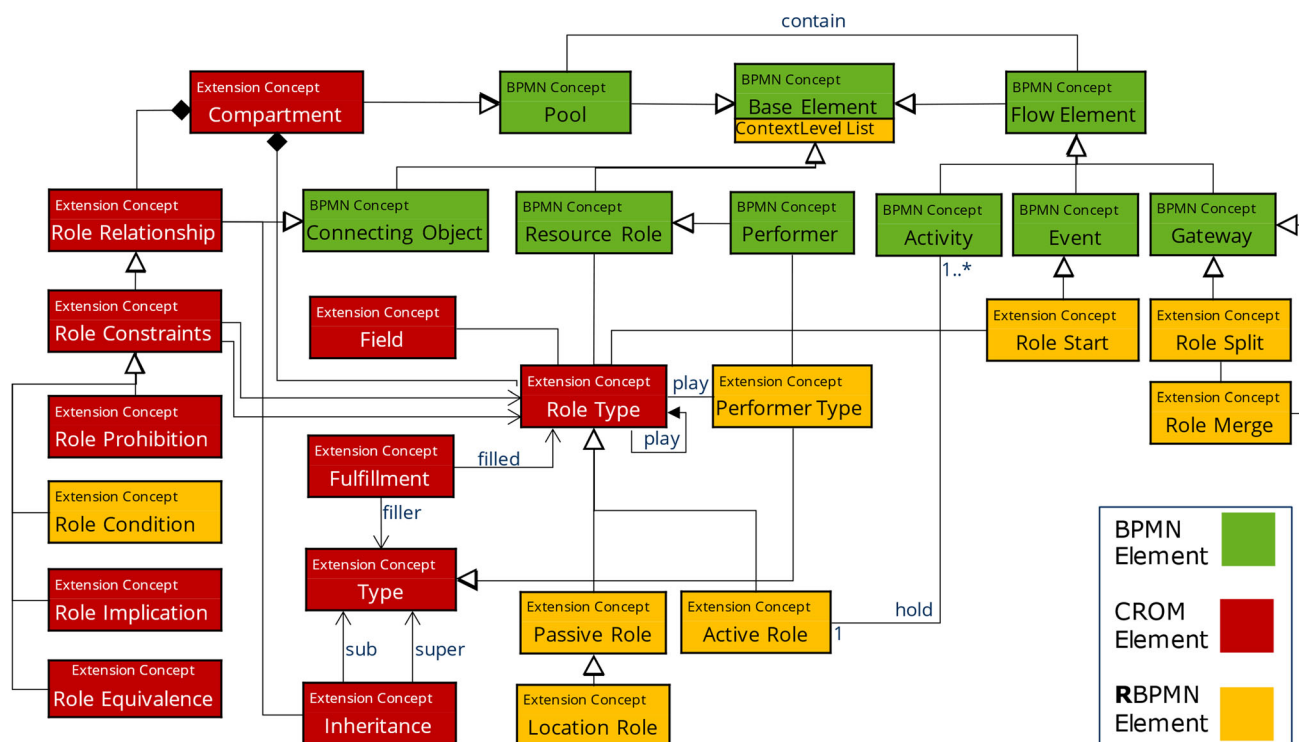
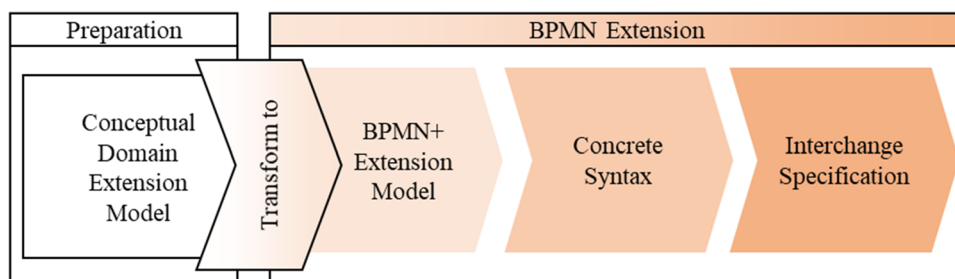


Fig. 6 The RBPMN metamodel as a UML class diagram

achieve O4. The active, passive, and location role were introduced to account for the variety of process performers in a modern process (see O1). Furthermore, roles have structural and behavioral aspects (see Sects. 3.1 and 4.1), which makes the role concept an ideal candidate to support the unification of these two perspectives (see O2). As [31] puts it (see Sect. 3.1), roles are fundamental to describing the world. This means they are a well-understood construct of the real world (O3). The introduction of role relationships was required to ease the unification of role-based behavioral and structural models (O2). To enhance the overall expressiveness of BP models, specifically for runtime resource allocation, the role start event and the Role Gateway were necessary (O1).

The role relationships role inheritance, role implication, and role prohibition were already established role relationships defined in the Compartment Role Object Model (CROM) [26]. Therefore, adhering to established role relationships made the introduction of those three necessary. The

role condition is a new relationship introduced to realize the workflow pattern Milestone [83] and improve resource allocation, as not every resource is needed at the start of the process and might never be needed.

5.5 Concrete syntax

Table 3 presents the concrete syntax of the RBPMN elements. As a BPMN extension needs to keep a BPMN model's look and feel [16], the new elements do not conflict with existing ones. Where not contradicting the extension restrictions of BPMN, we used the existing design of another role-based modeling language [28] for relations.

Because swimlanes are in their form rectangles, we decided on continuing the form of a rectangle for all roles, active, passive, and location. This decision was made following the “Physics of Notations” principle of perceptual discriminability, primarily visual distance and primacy of

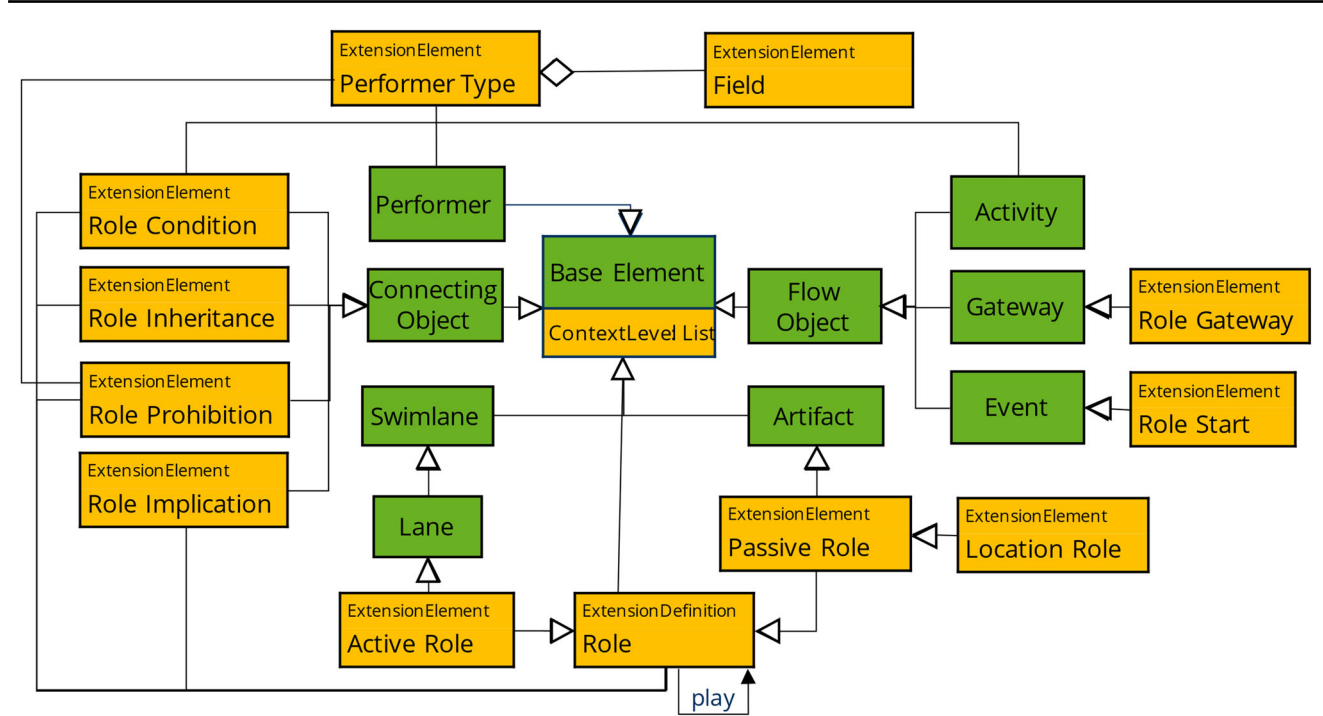


Fig. 7 The RBPMN Extension model as a UML class diagram

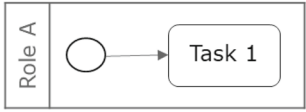
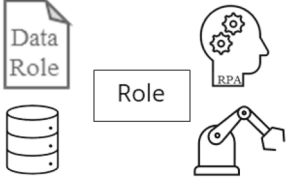
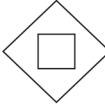
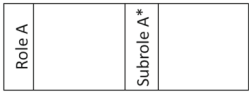
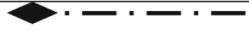
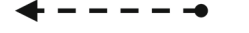
Table 1 BPMN + X transformation rules [82]

Class	Property	Type	Representation
BPMN concept	Original	Data type	BPMNElement property
BPMN concept	Original	BPMN concept	BPMNElement property
BPMN concept	New	Data type	ExtensionAttributeDefinition
BPMN concept	New	BPMN concept	ExtensionAttributeDefinition
BPMN concept	New	Extension concept	ExtensionAttributeDefinition
BPMN concept	New	Extension concept	ExtensionRelationship
Extension concept	New	Data type	Extensionelement Property/ExtensionAttributeDefinition
Extension concept	New	BPMN concept	Extensionelement Property/ExtensionAttributeDefinition
Extension concept	New	Extension concept	Extensionelement Property/ExtensionAttributeDefinition
Class	Generalization	Super class	Representation
BPMN concept	Original	BPMN concept	BPMNElement generalization
BPMN concept	New	BPMN concept	Invalid
BPMN concept	New	Extension concept	ExtensionRelationship
Extension concept	New	BPMN concept	ExtensionRelationship
Extension concept	New	Extension concept	ExtensionElement generalization
Extension concept	New	Extension concept	ExtensionElement generalization

Table 2 RBPMN element semantics

RBPMN element	Semantics
Active role	Where BPMN defines lane as a role, department, or system [9], RBPMN defines active role as a role with activities that change the state of something
Passive role	A new role type that RBPMN introduces. We differentiate role types because not every role which is necessary to achieve a business goal also performs or is responsible for a task. For example, the data role (in the BPMN data object) is important but has no tasks it is responsible for. A passive role can determine the state of something, e.g., a light sensor
Location role	A special passive role expressing a space (physical or virtual) where the tasks and roles it encapsulates in the model must be executed and played within said space. For example, a clean-air room in the semiconductor industry. The location role provides a state but does not change or determine the state of a role. It's a context provider, not a context modifier
Role start event	Denotes a role to be playable. It can be thrown by a task and is associated with the role condition relationship. It is used to specifically express the moment a performer starts playing a role in a BP
Role gateway	When splitting, it expresses one performer to now play multiple roles simultaneously. When merging, it expresses one performer drops all but one role
RBPMN relationship	Semantics
Role condition	Expresses that Task 1 must be executed for role A to be playable. For example, the task of defending the Ph.D. thesis must have happened to play the role of PostDoc
Role prohibition	Denotes that the same performer must not play Role A and Role B. For example, a performer must not be the Author and Reviewer of the same paper
Role inheritance	Expresses that role B inherits attributes and methods from role A. For example, the role Undergraduate Student inherits from the role Student
Role implication	Expresses that playing role A leads to playing role B. For example, playing the role Lecturer implies playing the role Faculty, but not all Faculty are Lecturers. The Lecturer does not inherit attributes from Faculty

Table 3 Concrete syntax of RBPMN elements

Element	Concrete Syntax
Active Role	
Passive Role	
Location Role	
Role Gateway	
Role Start Event	
Relationship	Concrete Syntax
Role Inheritance	
Role Condition	
Role Implication	
Role Prohibition	

shape [84]. Although the roles are different, they are closely related, and a model reader, even if unfamiliar, should be supported in recognizing the relationship.

RBPMN provides a generic construct for passive roles in the form of a rectangle. However, modelers are encouraged to use icons or pictograms where applicable. This distinction is made as symbolic signs allow for different interpretations of one element, which is also stated in The Physics of Notations' principle of semantic transparency [84]. In contrast, an iconic sign, like the pictogram used for a data role, limits the margin of misinterpretation [85]. This also relates to Challenge 1. Different process performer types can be expressed through different pictograms. The introduction of iconic signs for

yet-unknown process participants necessitates guidelines on how to use them. Iconic signs should only be used if the performer that will play this role is final at the design time, e.g., in our motivating example, multiple performers can play the role of color detector and therefore this role should be a rectangle, but only a camera can play the role of camera and therefore a camera pictogram can be used. There should only be one iconic sign of a performer type per model, ideally per organization. Performer types can differ in their granularity. A camera could be a security camera and a movie camera. In that case, the performers differ in their uses fundamentally, validating the use of different iconic signs.

The location role is a semi-transparent rectangle with double borders. As it overlays over tasks and can span multiple active roles, it is distinct from active and passive roles and other RBPMN elements.

Because we stayed with rectangles for roles, we also used rectangles for the role start event and role gateway. The circle shape of the event and the diamond shape of the gateway, as defined by the BPMN 2.0 standard, remain. Apart from the new elements, there are also new relationships added of which the role constraints were defined in [86].

The role inheritance between passive roles is a straight line with a white-filled arrowhead. This design is well known from the inheritance relationship in UML class diagrams [17]. The role implication is a dashed line with a circle at the currently played role and an arrowhead at the implied role. Designs for the role implication relationship from other role-based modeling languages such as BROS [36] or CROM [26] are equal or too similar to BPMN 2.0 relationships and, therefore, could not be chosen. A dashed line with a circle and an arrowhead does not yet exist in BPMN 2.0. Role condition is a dot-dash line with a diamond at the task and an arrowhead at the role. This is a new role relationship, and the dot-dash line does not exist yet in BPMN 2.0, making it easily distinguishable. Role prohibition is a dashed line with a backslash in a square in the middle of the path. The square containing the backslash was chosen, as it relates to the general prohibited sign.¹ Modelers are encouraged to place the square outside pools and lanes, to reduce cluttering of relationships in a model.

As Challenge 2 relates to the Physics of Notations principle of cognitive integration, the conceptual integration that RBPMN provides through the role concept was also intended to be highlighted in the concrete syntax [84]. However, as a BPMN 2.0 extension, it faces some restrictions that prohibit a higher cognitive integration by using the same concrete syntax of other role-based modeling languages [16].

5.6 Modeling example

To illustrate the use of the new RBPMN elements, we chose the Sorting Line process of the smart factory and added a quality control check to it (Fig. 8). The process starts when a thing playing the role of workpiece arrives at the sorting line station. A color detector is requested (triggering the start of the passive role color detector as shown through the role start events). Without the request, the color detector is not playable, as indicated through the Role Condition constraint. The workpiece is then moved to have its color determined through the Information System. The Mover role can be played by the Performers Conveyor Belt or Vacuum Gripper Crane. The possibility to detect and verify the color is provided through the location role Color Determinator. The Color Determinator is part of the Information System, which executes the tasks. The location role encapsulates the roles color detector and Color Verifier as well as the tasks Detect color and Verify color and finally determines the pickup area for the Colored Workpiece. The color is detected through the color detector role, which is played by the role of Color Sensor in the ideal process. The RGB-Camera (For which we chose the appropriate icon instead of the generic symbol), which plays the Surveillance Camera role, additionally starts playing the role of color verifier. This additional role is expressed through the role gateway. Once the color is verified, the Colored Workpiece role, inheriting from the Workpiece role provides information for the Information System, which pick up line to choose. Blue colored workpieces are moved to area 1, Red colored workpieces are moved to area 2, and White colored Workpieces are moved to area 3. Finally, the workpiece is released and from there, moved to the delivery and pickup station. It is important to ensure that color detector and color verifier are not played by the same performer, which we expressed through the role prohibition relationship between them.

If a human performer were to be considered in the example, they could play an active role (Mover or (new) Database-Editor) and a passive role (color detector or color verifier). They could detect the color of the workpiece and therefore determine the state and then enter the information into the database, thereby changing the state of the workpiece in the system.

6 Modeling context in RBPMN

Modeling is the act of a modeler expressing their perspective of a domain with a modeling language [87]. While we have stated that roles are context-sensitive (see Sects. 3.1 and 4.1), we have yet to show how context is modeled in RBPMN. We have introduced the modeling language in the

¹ ISO 3864-1.

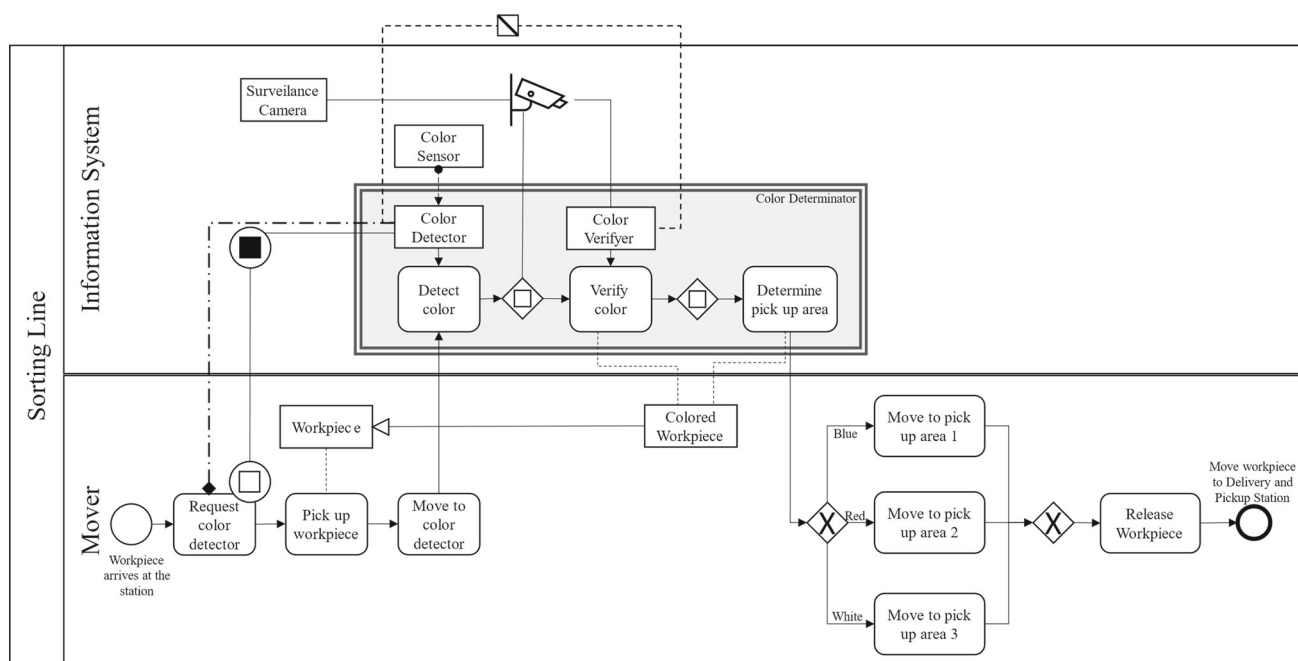


Fig. 8 RBPMN model of the sorting line process of the model smart Factory with a quality control check

previous section. This section introduces how context is modeled. We start by explaining how standard BPMN 2.0 models are designed.

6.1 Method and style vs. executable models

The BPMN modeling style differs for executable and non-executable process models. The non-executable process models are used by business users for the documentation and analysis of processes. The most prominent approach is “Method and Style” [19]. Following “Method and Style” results in well-structured processes that often hide the true complexity of activities. Complex user or machine tasks are displayed as atomic tasks, which is understandable for the modeler but unclear for a model reader unfamiliar with the process itself [19].

Executable process models, on the other hand, are typically more complex. They express data and dataflows explicitly. Compared to non-executable process models, the tasks are fine-grained, and the task types are well-defined. End events should always define distinct end states of the process and not generalize exceptions [19]. However, a harmonization of the two modeling styles never came to be. Rather, business users stick with “Method and Style” while the software developers create executable models. They are still modeled with one language but look very different.

Our focus is more toward executable process models that define tasks and events in a more fine-grained manner, for which the popular “Method and Style” only shows limited

applicability. Henceforth, we looked at existing modeling approaches that deal with context.

6.2 Context variability modeling

Our motivating example (Sect. 2.1, Figs. 1 and 2) shows that an executable model with some variability quickly results in a very large, complex, and hardly readable model. A different approach to modeling one large model that contains multiple variants is creating a set of models that share the same process (see Sect. 4.3). Creating multiple related models is useful when operating the same process in different contexts. Different contexts could be, for example, different countries, which require different regulations and, therefore, slight changes between the BPs. The models themselves remain smaller and more readable, but grasping the complexity of all variants combined is more difficult. Also, maintaining multiple models is more cumbersome.

One complex model is better for one context when the chosen variant is unclear until runtime [43]. Our motivating example is in the category of one context. Therefore, we chose the one-model approach to express variability. There is an objective context (workpiece, order type, environment status, etc.) of which parts are perceived through the available sensors. The executed process variant differs depending on how much of the objective context is perceived.

6.3 Role-based context modeling

In Sect. 3.1, we defined roles as contextual modeling constructs that can extend or change the specification and interactions of an object to fit the current context. Furthermore, roles can have or use sensory power, meaning they can perceive the context. Therefore, the subjectively perceived context defines the characteristics of a role.

In our motivating example, whether batch processing is made or a customized order can be fulfilled depends on the ability of the system to detect color. If the color sensor or the camera is available, the workpiece's color can be determined reliably. If only the brightness sensor is available, the determination of the color has a higher chance of error. The context of "determine color" objectively stays the same, as the workpiece does not change its color. However, the perception of the context is less rich in case of faulty sensors and, therefore, requires a higher adaptation. Adding a backup color sensor to the system could lead to a richer perception of the context and thus provide more possibilities to achieve the process goal.

Determining the specifications and interactions of a role within each subjectively perceived context requires every element of a BP to be provided with the context perceptions they appear in. Therefore, a new attribute `ContextLevel` is introduced to the BPMN 2.0 `BaseElement`. At the global level of the process, it contains an entry for every context perception and a list of all process elements within it. The list is filled through `ContextLevel` as an attribute of every process element. If any sensor fails, the BP flow must fall back to a BP variant that requires a less rich perception of the context. This is shown in our demonstration chapter in Figs. 11, 12 and 13.

6.4 Runtime monitoring of context

The chosen variant for the flow of a specific process instance is determined at runtime. While the variants for different levels of context perceptions are defined at design time, the chosen variant can only be determined once the context perception for this instance is final. This includes knowledge of all available sensors. Ensuring that the process goal can still be achieved even if a sensor fails requires changes of the variant at runtime. Of course, an additional benefit would be monitoring the current variant through the process model. This does not mean just a directly follows graph. It means reducing the possible process paths and updating the shown process model while it is executed.

This can be achieved through the variability concept hiding (Sect. 3.2). Suppose a sensor fails and the context perception is less rich. In that case, the process flow will drop to a context definition that does not require that sensor or the data provided by it. The process model is updated to display

the variant for the currently perceived context. The BP elements that are not defined for the updated context perception are hidden.

Especially in highly technical and complex processes, some later activities might rely on information generated early in the process. If the data generation failed earlier in the process, the later task that requires that data should no longer be a possible activity. The process execution should update to follow a variant that does not require these data or the activities. This process monitoring approach ensures that no activities are attempted later in the process that can only fail due to missing data.

The poorest context perception would be one that does not require additional data to achieve the process goal than what was available at the start of the process. However, suppose the context is perceived in its objective reality; in that case, the process flow can take different paths to create necessary data or estimate it well, even if one sensor fails.

7 Demonstration of context variability

We chose our motivating example to demonstrate how the requirements stemming from the smart factory's inbound delivery process are fulfilled.

We use color to highlight the different perceived contexts specified per process and attributes of the RBPMN elements. However, we assume that alternative color schemes or visualizations (e.g., filling patterns) could be used for the same purpose and should be implemented as user-based settings in respective modeling tools to accommodate for color-blind model readers, or readers from different cultural backgrounds [88]. In both examples, green shows the fail-safe context perception, where the process achieves the goal with as little sensor data as possible. Blue shows the maximum variability, where the process flow has many options to achieve the process goal because all intended sensors are available. The colors purple and orange show variants of perceived contexts between the two extremes. The context perceptions and, therefore, colors encompass the poor context perceptions. The context perception shown in blue encompasses all green, purple, or orange elements. Orange encompasses purple and green, and purple encompasses green. We show this explicitly in the Fischertechnik example (Sect. 2.1).

A quick recap of the Fischertechnik smart factory inbound delivery process that we explained as our motivational example (cf. Sect. 2.1): A workpiece's entry into the delivery and pickup station, signaled by interruption of the light barrier, initiates a system process to gather environment data and create a new product database entry. The vacuum gripper crane (VGR) retrieves the workpiece, and its color is identified via one of three sensor methods—a color sensor at the station,

an RGB camera, or a brightness sensor. The result of each method is recorded in the product database.

The workpiece is then moved to the NFC reader/writer, where existing information is read, and new data—delivery time, environment status, color, and product ID—is written. The workpiece is subsequently advanced to the next station.

If color detection sensors fail, preventing color determination, the process diverts from individual production to batch processing, given that color isn't crucial for the latter. The NFC reader/writer's role is considered non-essential for such cases, as it's primarily used for customer updates on customized products, which require specific color identification.

We modeled this process with the described process flow with its context variants in Fig. 9 in BPMN and Fig. 10 in RBPMN. At first glance, we can see that the RBPMN model is smaller because sensors are not modeled as lanes. Furthermore, dependencies throughout the process become clear in the RBPMN model, which is unclear in the BPMN model.

In Fig. 11, we see the poorest context perception that still achieves the process goal of moving the product to the next station (green). It still achieves this goal even if the connection to the information system is unavailable. The process can continue to provide full customization if the color sensor is available. This involves the elements that are highlighted green and purple (see Fig. 12). The color sensor could fail if the camera is available, and customization would still be available. This involves the elements that are green, purple, and orange (see Fig. 13). Two of the three color-determining sensors can fail without resorting to batch processing if the brightness sensor is available.

8 Evaluation

Evaluation of BP modeling languages is done either empirically or analytically. An empirical evaluation with users usually requires the modeling language to be well established and the users to have moderate experience with it [89]. This holds especially true when comparing a language extension to its source. As RBPMN is a very young language, this condition is not fulfilled. Henceforth, we decided to provide a thorough analytical evaluation in the form of user studies to make up for missing empirical evidence. An analytical evaluation is typically a comparison to a reference framework. In the context of our work, the most suitable approaches are the workflow patterns framework [83] and the ontology-based theory of representation [78, 90]. Role modeling provides an additional evaluation possibility in the form of role features [26]. Therefore, we present the evaluation of RBPMN for these three approaches. In addition, because our role-based context modeling approach (Sect. 6) addresses variability, we also evaluate RBPMN with the VIVACE framework [41].

8.1 Workflow patterns

Workflow patterns describe recurring problems and solutions of businesses. The more workflow patterns a BP modeling language can express, the more powerful it is [83]. The patterns can be categorized into different perspectives. The control flow perspective is the most relevant for evaluating BP modeling languages. In contrast, the other perspectives, resource, data, exception handling, presentation, and log imperfection, are more oriented toward evaluating tools [91]. As a BPMN standard conform extension, RBPMN can express all workflow patterns that BPMN can express. In addition, RBPMN can express the following workflow patterns: Milestone, Critical Section, Scope Data, Environment Data, Authorization, Separation of Duties, Retain familiar, Delegation, Stateful Reallocation, and Additional Resources.

The control-flow pattern Milestone describes that an activity can only be executed once a milestone is reached. If the milestone is not reached within a process instance, this activity can never be executed in this instance [92]. It can be expressed with the help of the role condition relationship. A role holding an activity can only be played once the milestone (a previous task or event) is executed. If the milestone is not achieved, the role is never played in this process instance. The Critical Section's control-flow pattern describes that activities in two parallel subgraphs can only be executed sequentially and never simultaneously, as they require the same resource [92]. It is expressed through the combination of role prohibition and role implication. The same performer must fulfill two parallel flows (role implication) but not concurrently (role prohibition). We modeled this pattern in Fig. 14. Role 1 and 3 must not be played in parallel by the same performer but are to be played by the same performer sequentially.

The data pattern Scope Data describes that a data resource can only be accessed by some tasks of a process, not by all tasks [92]. It can be expressed by modeling the data role inside a location role. This means that the scope of the data is limited to the location. In Fig. 14, only tasks A and B and the roles 1 and 2 have access to the data role P. The data pattern Environment Data describes that a data resource in the external operating environment of the process can be accessed during process execution [92]. It can be expressed through modeling a passive role outside a pool.

The resource pattern Authorization specifies a process performer's abilities [92]. It is expressed through the fact that the played role specifies the performer's behavior. The resource pattern Separation of Duties describes that different resources must execute two tasks for a process instance [92]. It can be expressed through role prohibition. This defines that the same performer must not play two roles.

In comparison with critical section, the role prohibition is modeled at the start of the roles. We show this pattern

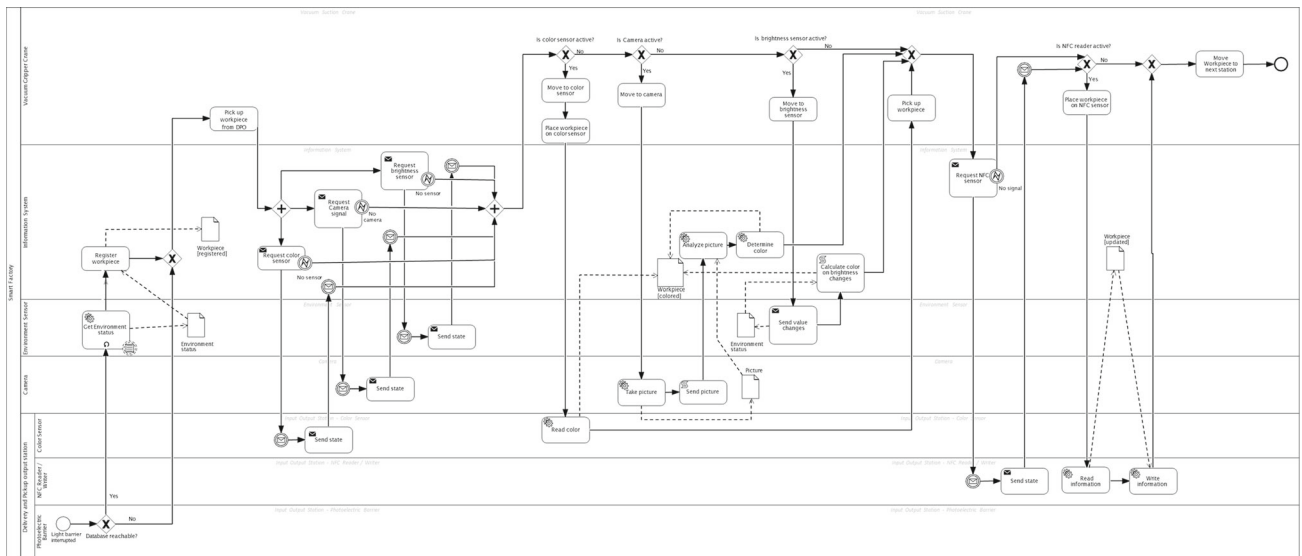


Fig. 9 BPMN 2.0 model of smart factory inbound delivery (zoom in the electronic version or find it at publisher's website)

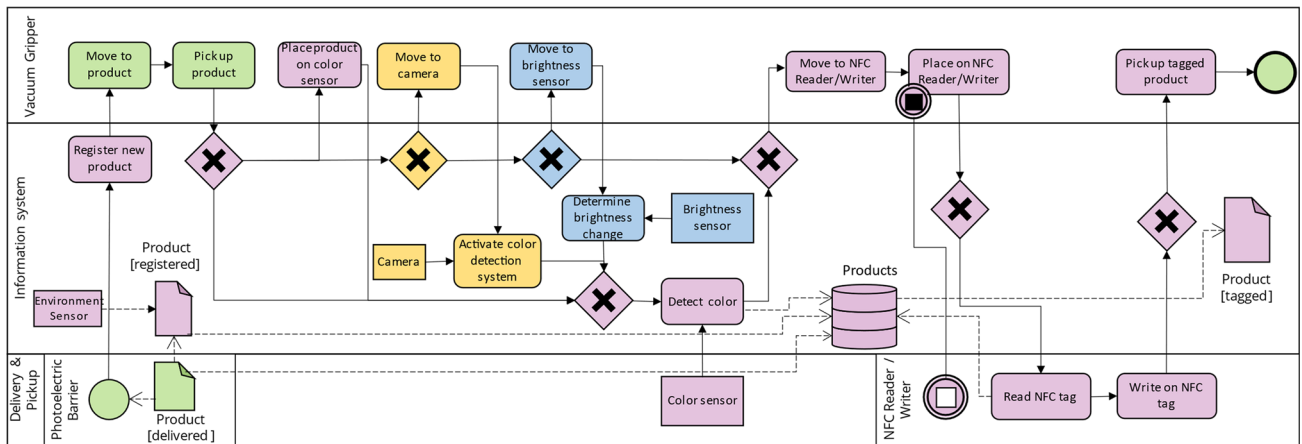


Fig. 10 RBPMN model of smart factory inbound delivery (zoom in the electronic version or find it at publisher's website)

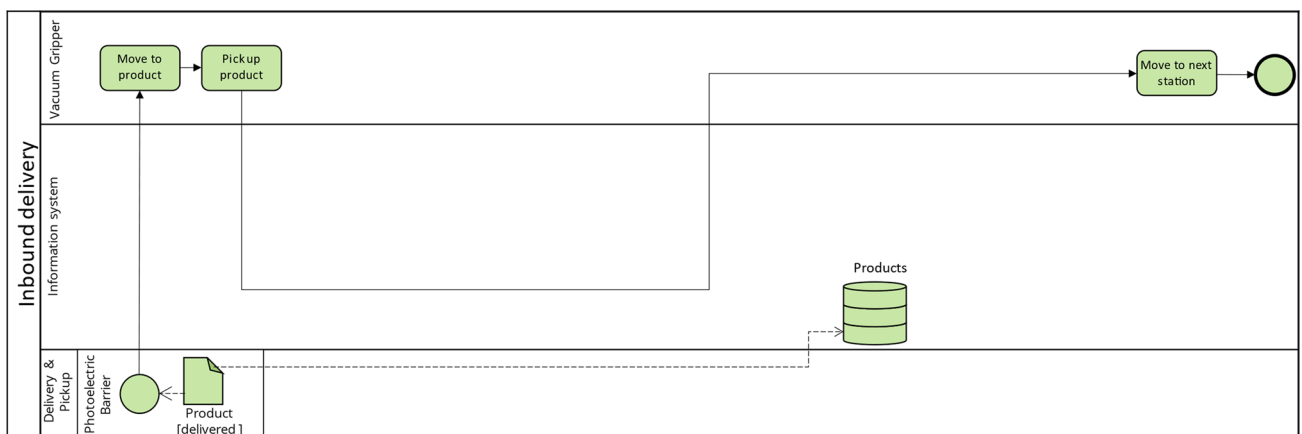


Fig. 11 Poor context perception of inbound delivery (zoom in the electronic version or find it at publisher's website)

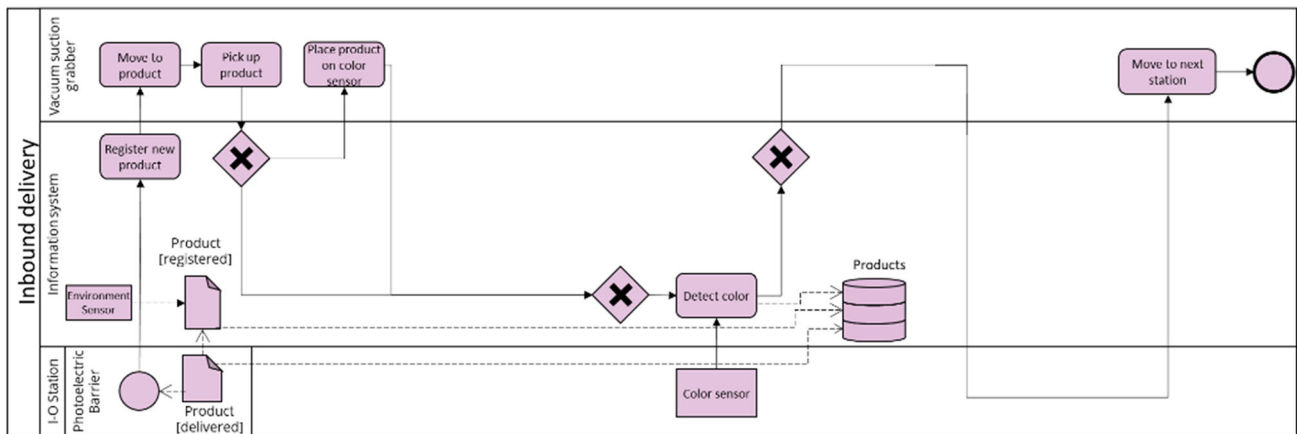


Fig. 12 Rich context perception inbound delivery (zoom in the electronic version or find it at publisher's website)

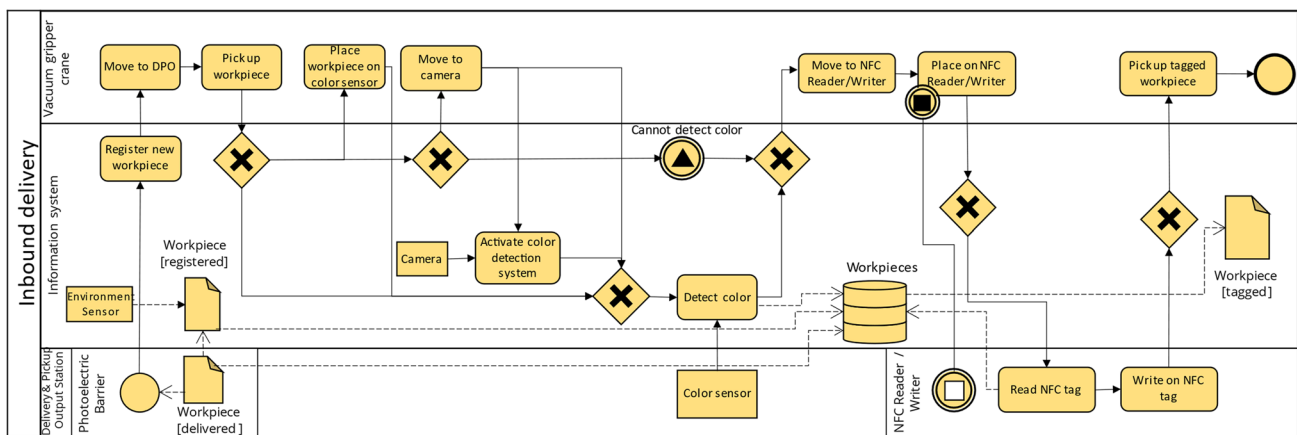
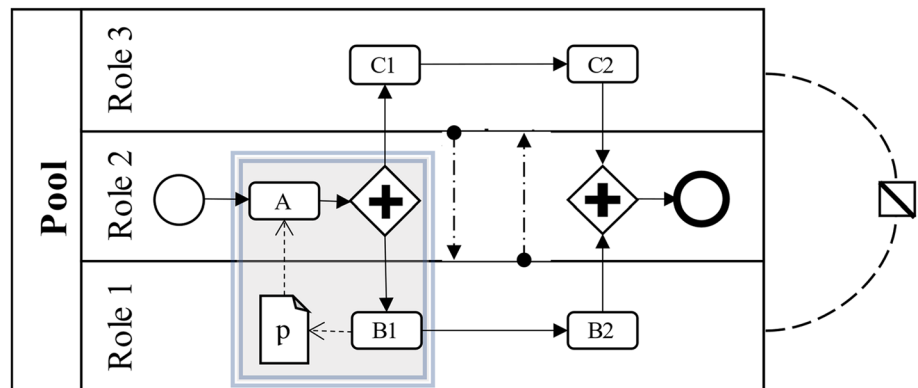


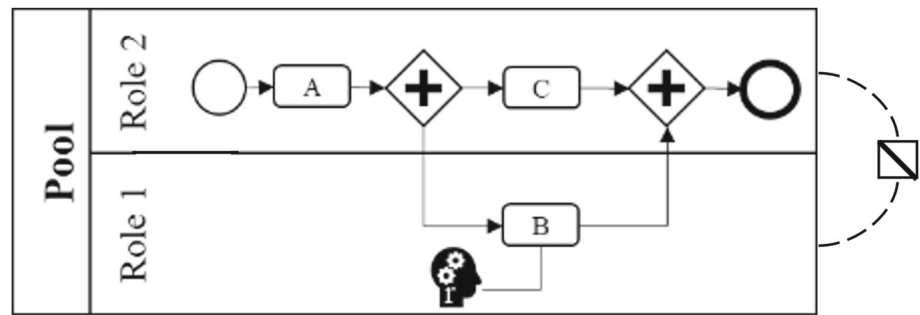
Fig. 13 Richer context perception inbound delivery (zoom in the electronic version or find it at publisher's website)

Fig. 14 Critical section and scope data



in Fig. 15. Retain Familiar's resource pattern describes that a resource that has previously handled a work item should continue to handle it [92]. It can be expressed through role implication. The resource pattern Delegation describes the possibility for a resource to delegate a work item to another resource [92]. It is expressed through the fact that the resource (performer) is not fixed to a role. Therefore, any other performer can play the role and execute the delegated task. The

resource pattern Stateful Reallocation [92] describes that a resource delegates a task to another resource while retaining the state of the work item. It is expressed through the fact that the relationships are between roles and not resources (performers). Therefore, another performer playing the same role will retain the state of the work item. The resource pattern Additional Resources [92] describes that a resource requests another resource to support the execution of a task. It can

Fig. 15 Workflow patterns separation of duties and additional resources**Table 4** Overview of workflow pattern coverage

Workflow pattern	Modeling language	
	BPMN [91]	RBPMN
Control-flow patterns	40/43	42/43
Data patterns	30/40	32/40
Resource patterns	31/43	37/43

be expressed through passive roles associated with a single task. The responsibility for the task remains with the role, the passive roles provide the support. In Fig. 15, the passive role *r* supports the task *B*.

Overall, RBPMN can express ten more workflow patterns than BPMN 2.0. We summarize this in Table 4. Therefore, it is a more powerful BP modeling language.

Naturally, the causes are additional elements compared to BPMN 2.0; hence, RBPMN is a more complex modeling language.

8.2 Ontology-based theory of representation

The dominant reference framework is the Bunge–Wand–Weber representation model (BWW) [90]. It is intended to describe every aspect of an information system. The evaluation of a language is based on two aspects: ontological completeness and ontological clarity.

Ontological completeness covers how many elements of the BWW are expressible in the modeling language. BPMN is already capable of expressing a large amount. Scholars have come to different conclusions about whether BPMN 2.0 can express State [93] or not [78]. We follow [93] that BPMN can represent state through the Data Object Reference [16]. RBPMN can express history additionally. History is “the chronologically ordered states a thing traverses” [90]. The role represents the state, and a performer’s roles are connected through role implication. Five elements of the BWW still remain that cannot be expressed. State Law, Conceivable Event Space, Stable State, Unstable State and Lawful Event Space. However, BWW was developed to represent a

complete information system, whereas our RBPMN presents only the behavior with structural hints. Therefore, a complete representation of the BWW was not the scope of RBPMN on its own.

Ontological clarity is measured through construct overload, redundancy, and excess [78, 89]. Construct overload occurs whenever a modeling construct describes more than one BWW element [78]. Construct redundancy occurs when one BWW element is described through multiple modeling language constructs [89]. Construct excess highlights modeling elements that are not relatable to any element of the BWW [89]. A standard conform extension only adds to the BPMN standard. Therefore, it cannot reduce construct excess or redundancy. Construct overload can, to some extent, be mitigated because lanes, now defined as roles, and pools, now understood as compartments, lose some of their ambiguity. Regarding ontological clarity, RBPMN does not perform significantly better than BPMN as seen in Table 5. The values for BPMN can be compared to [94].

8.3 Role patterns

The third evaluation we made is specifically for role-based modeling languages. Any role-based modeling language can be evaluated for its role-feature coverage of the 27 role features of Kühn [26]. RBPMN can express the following role features in addition to BPMN 2.0:

The sequence of role acquisition and removal may be restricted—The role implication relationship restricts the sequence in which roles are acquired. To further strengthen this, a combination of role implication and role prohibition provides a clear sequence restriction. It can potentially be argued that because a process is sequential, restrictions of the sequence occur naturally. We argue that this does not express a restriction, whereas our approach does. Roles can play roles is clearly defined in the metamodel of our approach and expressed through role inheritance in the model itself. Features of an object can be role-specific and are expressed in the metamodel as the role defines the object’s (performers’) properties. Different roles may share structure, and behavior can be expressed. Compartments may play roles that are

Table 5 Coverage of the Bunge–Wand–Weber representation model

Ontological construct	BPMN element	RBPMN element
Thing	Lane, pool	Role, pool
Properties	Attributes of pools, attributes of lanes	Attributes of roles, attributes of pools
State	Lane	Role
Conceivable state space	–	ContextPerceptionLevel
State law	–	–
Lawful state space	–	Context level list
Event	Start event, intermediate event, end event, message, timer, error, cancel, compensation, terminate, link, signal	Start event, intermediate event, end event, message, timer, error, cancel, compensation, terminate, link, signal
Event space	–	–
Transformation	Activity, task, collapsed sub-process, expanded sub-process, nested sub-process, transaction, call activity	Activity, task, collapsed sub-process, expanded sub-process, nested sub-process, transaction, call activity
Lawful transformation	Default flow, uncontrolled flow, exception flow	Default flow, uncontrolled flow, exception flow
Lawful event space	–	–
History	–	Role implication
Coupling	Message flow	Message flow
System	Pool, role	Pool, role
System composition	Pool, role	Pool, role
System environment	Pool, role	Pool, role
System structure	Pool, role	Pool, role
Subsystem	Pool, role	Pool, role
System decomposition	Pool, lane	Pool, lane
Level structure	Pool, role	Pool, role
Stable state	–	–
Unstable state	–	–
External event	Start event, intermediate event, end event, message, timer, error, cancel, compensation, terminate	Start event, intermediate event, end event, message, timer, error, cancel, compensation, terminate
Internal event	Start event, intermediate event, end event, message, error, cancel, compensation, signal, link, terminate	Start event, intermediate event, end event, message, error, cancel, compensation, signal, link, terminate
Well-defined event	Compensation, end event	Compensation, end event
Poorly defined event	Message, timer, error, cancel, terminate, start event, intermediate event	Message, timer, error, cancel, terminate, start event, intermediate event
Class	Lane, data object	Role
Kind	Lane	Subrole

part of themselves can be seen in subprocesses, where the pool (compartment) is split into multiple roles. The pool can play each of these roles. Different compartments may share structure, and behavior is defined in the metamodel.

We summarize the results in Table 6.

8.4 Process variability

The fourth evaluation is about variability modeling, which was not part of the original conference paper [7]. For this evaluation, we chose the VIVACE framework [41]. VIVACE is a framework for a systematic assessment of process variability approaches (shown in Table 7). A recent publication of it allows us to compare RBPMN with other BPMN 2.0 extensions that focus on variability [40]. The framework starts with defining the language. RBPMNs is a BPMN extension. Then, the technique of how the variability is modeled is either through a set of models (multi-artifact) or, as is the case for our RBPMN, within one model (single artifact). The next line is the method, explaining how variability is achieved. Our role-based context modeling approach is based on hiding process paths and elements. An element is hidden if a task, event, or role of its associated context perception fails. In the following section, we explain how RBPMN fulfills the variability-specific language constructs of Table 7.

8.4.1 Variability-specific language constructs

Elements within an RBPMN model possess the attribute `ContextLevel` (inherited from `BaseElement`) that defines their context perceptions, meaning certain process regions are hidden if the required context information is not perceived (C1 in Table 7). Which alternative is followed depends on the perceivable context information and what is defined as the standard path (C2 in Table 7). The context attribute defines what a configurable region or alternative requires to be available as a process path (C3 in Table 7). The chosen variant depends on the current context perception (e.g., Figs. 11 and 12, or 13). Any variant requiring currently not perceivable context information is unavailable as an alternative.

Furthermore, a new variant must contain the poorest context perception and be derived from the very rich context perception. This ensures that process model variants are always semantically and syntactically correct (C4 in Table 7). The variants are predefined at design time, but the one followed is only determined at runtime (C5 in Table 7).

8.4.2 Variability support features

Analysis and design phase We developed a modeling tool for RBPMN with the ADOxx development toolkit² to use

Table 6 Role feature coverage

Role feature	BPMN	RBPMN
1. Roles have properties and behaviors	●	●
2. Roles depend on relationships	●	●
3. Objects may play different roles simultaneously	●	●
4. Objects may play the same role (type) several times	◐	◐
5. Objects may acquire and abandon roles dynamically	◐	◐
6. The sequence of role acquisition and removal may be restricted	○	●
7. Unrelated objects can play the same role	○	●
8. Roles can play roles	◐	●
9. Roles can be transferred between objects	◐	◐
10. The state of an object can be role-specific	◐	◐
11. Features of an object can be role-specific	○	◐
12. Roles restrict access	◐	◐
13. Different roles may share structure and behavior	◐	●
14. An object and its roles share identity	◐	◐
15. An object and its roles have different identities	◐	◐
16. Relationships between roles can be constrained	○	○
17. There may be constraints between relationships	○	○
18. Roles can be grouped and constrained together	●	●
19. Roles depend on compartments	●	●
20. Compartments have properties and behaviors	●	●
21. A role can be part of several compartments	○	○
22. Compartments may play roles like objects	●	●
23. Compartments may play roles which are part of themselves	○	◐
24. Compartments can contain other compartments	●	●
25. Different compartments may share structure and behavior	○	◐
26. Compartments have their own identity	◐	◐
27. The number of roles occurring in a compartment can be constrained	●	●

No support: ○; partial support: ◐; full support: ●; only for instance level: ◑

² <https://www.adoxx.org/>.

Table 7 RBPMN evaluation regarding VIVACE-framework

Language	Extension	
Technique	Single-artifact	
Method	Hiding/blocking	
Perspectives	F, B, I, O	
Variability specific language constructs		
	C1 Configurable region	+
	C2 Configuration alternative	+
	C3 Configuration context condition	+
	C4 Configuration constraint	+
	C5 Configurable region resolution time	+
Variability support features	Analysis and design phase	
	F1.1 Modeling a configurable process model	+
	F1.2 Verifying a configurable process model and its related process family	+
	F1.3 Validating a configurable process model	+
	F1.4 Evaluating the similarity of different process variants	±
	F1.5 Merging process variants	+
	Configuration phase	
	F2 Configuring specific regions of a process variant out of a configurable process model	+
	Enactment phase	
	F3.1 Configuring specific regions of a process variant at enactment time	+
	F3.2 Dynamically configuring an instance of a process variant at enactment time	±
	Diagnosis phase	
	F4 Analyzing a collection of process variants	-
	Evolution phase	
	F5.1 Versioning of a configurable process model	+
	F5.2 Propagating changes of a configurable process model to already configured process variants	-
Tool implementation		M,G
Empirical evaluation		±
Application domain		CPS

the modeling language (F1.1). As there is a semantically and syntactically correct model for every context perception, verifying variants of a configurable model is possible (F1.2). Furthermore, every variant can be validated (F1.3). The similarity of BP variants can be compared by analyzing how much of a context perception is captured by another context perception. Therefore, an analysis of the intersection over union of two perceived contexts can evaluate the similarity between the two variants (F1.4). Because there are the richest and poorest perceptions of context, any process variant is between those two. A merge of multiple variants is limited to the richest perception context (F1.5). Therefore the analysis and design phase variability features [41] can be covered by RBPMN and our modeling tool.

Configuration phase The variant is configured by defining the context perceptions in which an RBPMN element is active or executed within the very rich context perception RBPMN model. Depending on the available context, information variants can be derived. The process variant of that context perception is displayed by setting a specific context perception to active within the tool. In contrast, anything outside this perception is hidden (F2).

Enactment phase During runtime, if required context information is unavailable (e.g., faulty sensor), the process can fall back to a variant requiring less context information (F3.1). This can continue until the point where the process follows the poorest perceivable context variant. It is just a switch from one predefined variant to another. Therefore, we conclude only partial fulfillment of (F3.2) as there is no move to a more perceived context for one instance. Some decisions, even though they would be available, are only decided during enactment time if certain context conditions are met.

Diagnosis phase There is no support for the Diagnosis phase (F4) as the VIVACE framework defines it [41]. The conditions for configuration at design and runtime are realized through the availability of contextual information. Suppose the information that is relevant to that variant is not present, then the variant chosen at design time is no longer followed at runtime.

Evolution phase The first variant and, therefore, the first version of a process should always be the one that requires the least amount of contextual information to achieve the process goal. Any addition of contextual information, for example, through additional sensors, results in a new context perception and, therefore, a new version of the process. The very rich context perception is extended each time a new sensor or actuator is added and becomes the next version of the process model (F5.1). RBPMN, with our modeling style, does not support the automatic propagation of changes to one context

perception. One could argue that any change to the poorest perceivable context process model is directly translated to all other models. However, changes to the very rich perceivable context process model (the configurable model) are not translated to variants if the changes only concern elements outside the context perception of the variants (F5.2).

Tool implementation For tool implementation (see Table 7), we can thus far say that there is a tool for the language. It enables the design of a configurable process model, but the tool is not publicly available yet. There is support for modeling and configuring variants, but no guidelines for deriving process variants exist.

Empirical evaluation The VIVACE framework also addresses empirical evaluation. We do not see the aspect of empirical evaluation as completely fulfilled because even though we showed its applicability through the demonstration and the different evaluations, user studies are missing. These remain open, and we intend to conduct them soon.

Application domain The application domain (Table 7 last line) we show in this paper is CPS. However, RBPMN is intended as a domain-independent modeling language.

RBPMN can provide a high degree of variability based on our evaluation through the VIVACE framework. However, RBPMN does not incorporate the temporal or operational perspective. Furthermore, the empirical evaluation through user studies is missing, the tool is not published with a proper manual for general use, and variant diagnosis and propagation of changes are lacking.

8.5 Physics of notation

We do not claim that a theoretical evaluation of the comprehensiveness of RBPMN is comparable to results from a user study. Nevertheless, we can argue that Daniel Moody's Physics of Notation provides a well-established framework for the development and evaluation of modeling languages [84]. Our evaluation is centered on the extent to which the nine design principles are successfully integrated in RBPMN. By comparing our findings to those of the widely used BPMN 2.0 standard [95], we provide a comprehensive assessment of RBPMN's capabilities.

1. **Principle of Semiotic Clarity:** There should be a 1:1 correspondence between semantic constructs and graphical symbols [84].

This principle analyzes symbol redundancy, symbol overload, symbol excess, and symbol deficit. These aspects were already evaluated exhaustively in Sect. 8.2. As a standard-conformant BPMN extension, RBPMN can alleviate symbol

deficit by introducing new elements, whereas it cannot mitigate symbol excess.

2. Principle of Perceptual Discriminability: Different Symbols Should Be Clearly Distinguishable from Each Other [84].

In RBPMN, all roles are represented as rectangles, which may compromise discriminability. However, active roles can be distinguished by containing activities, and location roles are visually distinct due to their double border and semi-transparent design, thus preserving some level of discriminability. Furthermore, the use of iconic signs for passive roles can enhance discriminability. The relationships in RBPMN that diverge from those in BPMN are differentiated through distinct arrowheads and line styles. Notably, RBPMN, similar to BPMN, does not employ textual differentiation [16, 95].

3. Principle of Semantic Transparency: Use Visual Representations Whose Appearance Suggests Their Meaning [84].

In contrast to BPMN, RBPMN suggests the adoption of iconic signs to enhance visual representation. BPMN already employs iconic signs for Data Object, Data Storage, and Message [16]. Furthermore, RBPMN allows for the use of iconic signs for each passive role, thereby increasing the semantic transparency of the model. This advancement enables a more intuitive understanding of the model's structure and relationships.

4. Principle of Complexity Management: Include Explicit Mechanisms for Dealing with Complexity [84].

The fourth principle of the Physics of Notation highlights the importance of facilitating the decomposition of complex processes into more manageable parts. In RBPMN, the role-based approach effectively achieves this by introducing a clear hierarchical structure and sequence through the definition of roles and role relationships. The role gateway and role start event mechanisms further enable the management of role transitions, allowing for a more controlled and efficient process flow.

5. Principle of Cognitive Integration: Include Explicit Mechanisms to Support Integration of Information from Different Diagrams [84].

The fifth principle of the Physics of Notation emphasizes the importance of providing a coherent and unified view of complex systems by integrating multiple process models. BPMN partially addresses this principle through

subprocesses and link events, enabling the explanation of how multiple process models relate to each other and form a larger system [95]. RBPMN takes a significant step further by proposing a unified modeling concept that bridges the gap between structural and behavioral modeling [96]. This unified approach enables better cognitive integration, as it not only integrates diagrams of the same type but also provides a holistic view of the system.

6. Principle of Visual Expressiveness: Use the Full Range and Capacities of Visual Variables [84].

Since RBPMN allows tasks to involve multiple roles and therefore span multiple lanes, the size of a task is meaningful. Hence, task size facilitates visual expressiveness that goes beyond what BPMN can provide [95].

7. Principle of Dual Coding: Use Text to Complement Graphics [84].

For BPMN and RBPMN text is not an adequate option for encoding information [95]. Therefore, both notations fail to make good use of this principle. Dual Coding is primarily used for gateways to describe the conditions.

8. Principle of Graphic Economy: The Number of Different Graphical Symbols Should Be Cognitively Manageable [84].

As a standard-conform BPMN extension, RBPMN cannot improve graphic economy since it includes all BPMN elements. The principle of graphic economy suggests a maximum of six different graphical constructs for novices [84]. BPMN has 171 [95], and RBPMN introduces even more with the possibility of using iconic signs. Both modeling languages perform poorly in this regard.

9. Principle of Cognitive Fit: Use Different Visual Dialects for Different Tasks and Audiences [84].

Elements from other modeling languages, such as UML class diagrams and business role-object specification, can enhance cognitive fit by leveraging users' existing knowledge and familiarity. The concept of roles, which is fundamental to our daily lives, can also contribute to improved cognitive fit. However, for users without prior knowledge of the role concept presented in RBPMN, the cognitive fit may be compromised due to the introduction of an additional unfamiliar concept. Regarding the idea of different visual dialects, RBPMN aims to bridge this gap by providing a common visual language that can facilitate communication and understanding among different stakeholders. In this context, BPMN, as a standardized notation, can offer a greater

cognitive fit when used in conjunction with different visual dialects. (e.g., Method and Style for analytics and executable models for programming).

Overall, RBPMN adheres stricter to the Physics of Notation for the principles of semiotic clarity, semantic transparency, cognitive integration, and visual expressiveness. RBPMN, compared to BPMN, adheres equally well to the Physics of Notation for the principles of perceptual discriminability, cognitive fit, dual coding, and complexity management. Compared to BPMN, RBPMN performs worse in terms of graphical economy. This is due to the introduction of additional graphical constructs, such as iconic signs, which can increase the visual complexity of the notation.

9 Discussion

In line with the design science research methodology, we started the development of our solution to address the challenges that arose from our example (see Sect. 2.1) and from researchers (see Sect. 2.2). We formulated our research questions and objectives based on these challenges. We will provide the answers to the questions and how the objectives are fulfilled in the following subsections.

9.1 Objective 1: increase the expressiveness of BP models to model a variety of process participants

The process performers are sorted into active roles that change the state of others and passive roles that determine the state of others (see Sect. 5). The type of performer, be it human, machine, sensor, actuator, system, or anything else, is not influencing the determination of the role type they play. The intended abilities define if a performer plays an active or passive role in a process. The role types are modeled differently because, if passive roles were modeled as swimlanes, like the sensors were in our motivating example (Sect. 2.1, Figs. 1 and 2), the number of swimlanes in a process explode. Overall, our RBPMN can cope with the variety of process performers and provide a concept to model new process performer types that are not thought about yet. Therefore, we consider this objective fulfilled.

9.2 Objective 2: increase the overlap of the structural and the behavioral modeling perspectives

Implementing a BP requires stakeholders of different backgrounds who use different constructs to address different aspects of the process to communicate. Suppose the stakeholders can spot where their aspects are considered in another stakeholders' model. In that case, they are assured there is

no ambiguity regarding the BP. RBPMN has roles as the key modeling construct and role relationships for modeling some structural aspects. Roles have structure and behavior (see Sect. 8.3, features 1 and 13). With the role inheritance relationship, we make relations between roles more explicit. The BPMN 2.0 standard only implies inheritance between data objects through naming. RBPMN enables modelers to make it explicit so that there is no confusion from which role another role (or specifically data role, for that matter) inherits attributes.

RBPMN was designed with the behavior-aware structural modeling language BROS [36] as the complement in mind. Both modeling languages share a coherent role understanding, which enables a truthful translation between structural and behavioral models [96]. Overall, we see this objective as mostly fulfilled. While a BP model could have more structural aspects, this would conflict with objective 4.

9.3 Objective 3: increase comprehensiveness of BP models through modeling constructs that are well understood in the real world

In Sect. 3.1, we cited Steimann, who argued that roles are fundamental to understanding the world [21]. Our goal was to use this well-known concept from the real world, rather than a more abstract concept like class, to make business process (BP) models easier to understand. We wanted to improve comprehension of BP models, not just simplify them or the modeling languages used to create them. There are already BP modeling languages that have relatively few objects and relations, like the UML Activity Diagram [12], compared to the more complex BPMN 2.0 [4]. However, using a complex language does not necessarily result in complex models, as more aspects can be explicitly expressed. Our RBPMN introduces new modeling constructs (see Sect. 5.3ff), which increases the complexity of the modeling language compared to BPMN 2.0.

While the complexity has increased, it could be assumed that understanding RBPMN models from a structural perspective might be simpler than BPMN, given that a number of the new constructs have a direct translation into structural modeling with BROS [96]. In addition, as our evaluations in Sects. 8.1 and 8.2 demonstrate, RBPMN is more expressive than BPMN. The new RBPMN constructs not only unify behavioral and structural modeling but also increase expressiveness. We feel that we might have achieved our objective to some extent. RBPMN models could potentially be more comprehensive for stakeholders familiar with the role concept, and it's realistic to assume that their ontological clarity has not been reduced (Sect. 8.2). Our context modeling approach might assist model readers in understanding how multiple BP model variants are connected (Sect. 6). We made a first assessment using the Physics of Notation framework of [84]

as previously done by [95] and compared RBPMN to their evaluation of BPMN. However, a fully objective assessment of the comprehensiveness of RBPMN would necessitate user studies with novices and experts in the language, which have not been conducted yet.

9.4 Objective 4: enhance variability modeling for context-sensitive processes

Our context modeling approach (see Sect. 6) supports modelers in expressing process variants for different levels of context perception. Our evaluation through the VIVACE framework [31] (Sect. 8.4) showed that RBPMN can support most variability concepts. Compared to other BPMN extensions that were evaluated by the VIVACE framework in a recent literature review [40], RBPMN supports more variability, although using context parameters to define variant choice is not a new approach. However, the variability modeling in RBPMN is for different context perceptions, not for different contexts.

Variability is just one of four aspects of business process flexibility [43]. This paper did not address the other aspects: adaptation, looseness, and evolution. Furthermore, our approach to model context-sensitive variability was demonstrated with one CPS example. While we assume that our findings and the benefits of our approach hold for other domains, proving it was out of the scope of this paper. Nevertheless, RBPMN clearly enhances variability modeling for context-sensitive processes. This is why we consider this objective satisfied.

Overall, the research objectives O1, O2, O3, and O4 were fulfilled to a high degree. This also enables us to answer our research questions. The answer to the first research question: “How can the expressiveness, variability, and comprehensiveness of business process models be improved?” is through incorporating a modern understanding of roles (Sects. 5 and 6). The answer to the second research question: “How can context-sensitive variability of business process models be improved?” is through the definition of different levels of context perception for context-sensitive process elements (Sect. 6).

However, there are some limitations of our work we must address. RBPMN is not a cure-all to every modern BP modeling challenge (Sect. 2.2). Unifying behavior modeling with other modeling perspectives requires the coherent use of roles in all perspectives. RBPMN supports this in combination with BROS [36, 96], but there are more perspectives to systems. The value of roles for perspective unification drops if only one language uses them without a coherent understanding. Loose and evolving processes require continuous structural changes even at runtime [43], which is something neither RBPMN nor our context modeling approach can freely express.

Had we started our research endeavor with a different role understanding or developed an entirely new BP modeling language, potentially all objectives could have been fulfilled completely, or we could have addressed more challenges. But, as we explained in Sect. 5.1, the downsides of developing a new language led us to extend an existing standard.

RBPMN is thoroughly evaluated from an analytical point of view and has proven its value and the value of roles. The resulting RBPMN models of one CPS are more expressive than the BPMN 2.0 models and can contain multiple variants that are based on different levels of context perception.

10 Conclusion and future work

BP modeling is in a crisis because modern BPs have become increasingly complex. Modern BPs are context-sensitive interactions of diverse process performers, such as humans, machines, systems, sensors, actuators, software, and more (Challenge 1). The corresponding BP models must represent their implementation truthfully (Challenge 2) to ensure that the required context-sensitivity is upheld (Challenge 4) and that automation of the BPs is possible. Of course, as our motivating example had shown, even small BPs result in complex models that are hard to comprehend (Challenge 3). Context sensitivity also leads to a large variety of BP models, which puts additional pressure on variability requirements (Challenge 5).

We contribute RBPMN, a standard-conform BPMN extension, and a context-sensitive modeling approach to overcome these challenges. Challenge 1 is solved with the active, passive, and location roles of RBPMN. The performer type, be it human, machine, or else, is irrelevant to the BP, as the grouping is based on the played role. Challenge 2 is addressed through the role relationships, which express structural aspects hidden in BPMN 2.0, and roles as the key modeling construct because roles are found in many stakeholder perspectives. Challenge 3, increasing the comprehensiveness of BP models, was approached through roles, as they are a well-known concept in the real world. However, no concluding statement can be made on whether this challenge has been solved or not until empirical evidence is gathered in future work.

Furthermore, adding structural aspects for challenge 2 also enables stakeholders from the structural perspective to better understand an RBPMN model. Roles are context-sensitive by their nature (see Sect. 3.1), which inspired us to extend BPMN 2.0 to introduce a context attribute that is based on context perception to solve challenge 4. Every BP modeling element possesses it (Sect. 6), which defines at which level of context perception an element becomes active in a process. This also helps for challenge 5, for which RBPMN can reduce the complexity of multiple BP variants within one context.

This can also enable analysts to monitor the current active variant of a BP at runtime (Sect. 6.4).

A high degree of variability can be expressed with our RBPMN. We demonstrated its potential with an exemplary model of a smart factory. Our evaluation of RBPMN through workflow patterns (Sect. 8.1), representation theory (Sect. 8.2), role feature coverage (Sect. 8.3), VIVACE (Sect. 8.4), and the Physics of Notation framework (Sect. 8.5) show the value a role-based approach has for BP modeling.

Overall, our evaluation shows that RBPMN is a promising approach that can solve the challenges of modern business process modeling. Practitioners benefit from easier unification with other modeling perspectives, from modeling with a well-understood construct, and from the improved context-sensitive variability of their BP models. The value of roles for business process modeling could only be greater if complementary models of other perspectives are role-based as well.

Future work will conduct a user study to evaluate the comprehensiveness of RBPMN for novices of the modeling language. The semantics of RBPMN elements were designed to fulfill the research objectives. An improvement in future work will be to ground them on a foundational ontology. Furthermore, we will conduct a case study with two large processes from the banking domain to investigate if the benefits stay true in other domains. A fundamental aspect of this study is the validation of our approach for semi-automated processes where human involvement is crucial. This necessitates a more sophisticated context modeling strategy, as humans can be viewed as multi-sensor entities that can alter the state of a process. As a result, one individual can assume both active and passive roles within the same process instance, adding complexity to the process model. In addition, the runtime support (Sect. 6.4) works best in a role-based environment, which is not the current standard. Henceforth, we will extend our RBPMN modeling tool and provide it to the public.

Acknowledgements This work is partially funded by the German Research Foundation (DFG) within the Research Training Group “Role-based Software Infrastructures for continuous-context-sensitive Systems” (GRK 1907). We thank Prof. Dr. Barbara Weber for her valuable input.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your

intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

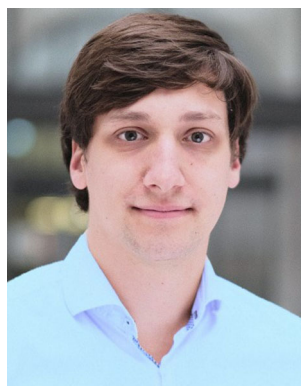
1. Kannengiesser, U., Müller, H.: Subject-Oriented for Human-Centred Production: A Research Agenda. In: Fischer, H., Schneeberger, J. (eds.) *S-BPM ONE—Running Processes*, pp. 235–244. Springer, Berlin Heidelberg (2013)
2. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., Fortino, G., Gal, A., Kannengiesser, U., Leotta, F., Mannhardt, F., Marrella, A., Mendling, J., Oberweis, A., Reichert, M., Rinderle-Ma, S., Serral, E., Song, W., Su, J., Torres, V., Weidlich, M., Weske, M., Zhang, L.: The internet of things meets business process management: a manifesto. *IEEE Syst. Man Cybern. Mag.* **6**, 34–44 (2020)
3. Seiger, R., Zerbato, F., Burattin, A., García-Bañuelos, L., Weber, B.: Towards IoT-driven Process Event Log Generation for Conformance Checking in Smart Factories. In: *2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pp. 20–26 (2020)
4. Seiger, R., Franceschetti, M., Weber, B.: An interactive method for detection of process activity executions from IoT data. *Future Internet*. **15**, 77 (2023)
5. Torres, V., Serral, E., Valderas, P., Pelechano, V., Grefen, P.: Modeling of IoT devices in Business Processes: A Systematic Mapping Study. In: *2020 IEEE 22nd Conference on Business Informatics (CBI)*, pp. 221–230 (2020)
6. Seiger, R., Keller, C., Niebling, F., Schlegel, T.: Modelling complex and flexible processes for smart cyber-physical environments. *J. Comput. Sci.* **10**, 137–148 (2015)
7. Skouti, T., Furrer, F.J., Strahringer, S.: RBPMN—The Power of Roles for Business Process Modeling with BPMN. In: Augusto, A., Gill, A., Bork, D., Nurcan, S., Reinhartz-Berger, I., Schmidt, R. (eds.) *Enterprise, Business-Process and Information Systems Modeling*, pp. 29–42. Springer International Publishing, Cham (2022)
8. Silver, B.: *BPMN Method and Style: with BPMN Implementer’s Guide*. Cody-Cassidy Press, Aptos (2011)
9. Weber, M., Grisold, T., vom Brocke, J., Kamm, M.: Context-aware business process modeling: Empirical insights from a project with a globally operating company. In: *ECIS 2021 Research-in-Progress Papers*, p. 13 (2021)
10. Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: A systematic literature review on IoT-aware business process modeling views, requirements and notations. *Softw. Syst. Model.* (2022)
11. Hasic, F., Asensio, E.S.: Executing IoT Processes in BPMN 2.0: Current Support and Remaining Challenges. In: *2019 13th International Conference on Research Challenges in Information Science (RCIS)*, pp. 1–6. IEEE, Brussels, Belgium (2019)
12. Beverungen, D., Buijs, J.C.A.M., Becker, J., Di Ciccio, C., van der Aalst, W.M.P., Bartelheimer, C., vom Brocke, J., Comuzzi, M., Kraume, K., Leopold, H., Matzner, M., Mendling, J., Ogonek, N., Post, T., Resinas, M., Revoredo, K., Río-Ortega, A., La Rosa, M., Santoro, F.M., Solti, A., Song, M., Stein, A., Stierle, M., Wolf, V.: Seven paradoxes of business process management in a hyper-connected world. *Bus. Inf. Syst. Eng.* **63**, 145–156 (2021)
13. Dumas, M., Fournier, F., Limonad, L., Marrella, A., Montali, M., Rehse, J.-R., Accorsi, R., Calvanese, D., De Giacomo, G., Fahland, D., Gal, A., La Rosa, M., Völzer, H., Weber, I.: *Augmented Business Process Management Systems: A Research Manifesto*. ArXiv220112855 Cs. (2022)

14. Beerepoot, I., Di Ciccio, C., Reijers, H.A., Rinderle-Ma, S., Bandara, W., Burattin, A., Calvanese, D., Chen, T., Cohen, I., Depaire, B., Di Federico, G., Dumas, M., van Dun, C., Fehrer, T., Fischer, D.A., Gal, A., Indulska, M., Isahagian, V., Klinkmüller, C., Kratsch, W., Leopold, H., Van Looy, A., Lopez, H., Lukumbuzya, S., Mendling, J., Meyers, L., Moder, L., Montali, M., Muthusamy, V., Reichert, M., Rizk, Y., Rosemann, M., Röglinger, M., Sadiq, S., Seiger, R., Slaats, T., Simkus, M., Someh, I.A., Weber, B., Weber, I., Weske, M., Zerbato, F.: The biggest business process management problems to solve before we die. *Comput. Ind.* **146**, 103837 (2023)
15. Ozkaya, M., Erata, F.: Understanding practitioners' challenges on software modeling: a survey. *J. Comput. Lang.* **58**, 100963 (2020)
16. Object Management Group (OMG) ed: Business Process Model and Notation (BPMN), Version 2.0. <https://www.omg.org/spec/BPMN/2.0> (2011)
17. Object Management Group (OMG): Unified Modeling Language, v2.5.1. <https://www.omg.org/spec/UML> (2017)
18. Pereira, J.L., Silva, D.: Business Process Modeling Languages: A Comparative Framework. In: Rocha, Á., Correia, A.M., Adeli, H., Reis, L.P., Mendonça Teixeira, M. (eds.) *New Advances in Information Systems and Technologies*, pp. 619–628. Springer International Publishing, Cham (2016)
19. Silver, B.: Executable BPMN vs Method and Style, <https://methodandstyle.com/executable-bpmn-vs-method-and-style/> (2022). Accessed 15 Nov 2022
20. Sokolowski, D., Weisenburger, P., Salvaneschi, G.: Change is the only constant: dynamic updates for workflows. In: *Proceedings of the 44th International Conference on Software Engineering*, pp. 350–362. ACM, Pittsburgh Pennsylvania (2022)
21. Marrella, A., Mecella, M., Sardina, S.: Intelligent process adaptation in the SmartPM system. *ACM Trans. Intell. Syst. Technol.* **8**, 1–43 (2017)
22. Seiger, R., Huber, S., Heisig, P., Aßmann, U.: Toward a framework for self-adaptive workflows in cyber-physical systems. *Softw. Syst. Model.* **18**, 1117–1134 (2019)
23. De Luzi, F., Leotta, F., Marrella, A., Mecella, M.: On the interplay between business process management and internet-of-things: a systematic literature review. *Bus. Inf. Syst. Eng.* (2024)
24. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**, 75–105 (2004)
25. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **24**, 45–77 (2007)
26. Kühn, T.: A Family of Role-Based Languages. <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-228027> (2017)
27. Kühn, T., Werner, C., Schon, H., Zhenxi, Z., Aßmann, U.: Contextual and Relational Role-Based Modeling Framework. In: *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 442–449. IEEE, Kallithea-Chalkidiki, Greece (2019)
28. Schön, H., Strahringer, S., Furrer, F.J., Kühn, T.: Business Role-Object Specification: A Language for Behavior-aware Structural Modeling of Business Objects. In: Ludwig, T., Pipek, V. (eds.) *Wirtschaftsinformatik 2019 Proceedings*, pp. 244–258 (2019)
29. Kluge, T.: A role-based architecture for self-adaptive cyber-physical systems. *Phys. Syst.* **5** (2020)
30. Bachman, C.W., Daya, M.: The role concept in data models. *Proc. Third Int. Conf. Very Large Data Bases* **3**, 464–476 (1977)
31. Steimann, F.: The role data model revisited. *Appl. Ontol.* **2**, 89–103 (2007)
32. Saidani, O., Nurcan, S.: A Role-Based Approach for Modelling Flexible Business Processes. *Proc. CAISE06 Workshop Bus. Process Model. Dev. Support.*, pp. 111–120 (2006)
33. Balabko, P., Wegmann, A., Ruppen, A., Clément, N.: The Value of Roles in Modeling Business Processes. In: *Proceedings of the 5th Workshop on Business Process Modeling, Development and Support (BPMDS 2004)*, p. 9. Riga (2004)
34. Caetano, A., Silva, A.R., Tribolet, J.: Using roles and business objects to model and understand business processes. In: *Proceedings of the 2005 ACM symposium on Applied computing—SAC '05*, p. 1308. ACM Press, Santa Fe, New Mexico (2005)
35. Kim, D.-K., Chung, Y.K.: R-BPMN for abstract modeling of business process patterns. *Bus. Process. Manag. J.* **27**, 1445–1462 (2021)
36. Schön, H.: Role-based Adaptation of Business Reference Models to Application Models. <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-725441> (2020)
37. Kühn, T., Leuthäuser, M., Götz, S., Seidl, C., Aßmann, U.: A metamodel family for role-based modeling and programming languages. In: *Software Language Engineering*, pp. 141–160. Springer (2014)
38. Bera, P., Burton-Jones, A., Wand, Y.: Improving the representation of roles in conceptual modeling: theory, method, and evidence. *Requir. Eng.* **23**, 465–491 (2018)
39. Ferraiolo, D.F., Kuhn, D.R.: Role-Based Access Controls. *15th Natl. Comput. Secur. Conf.*, pp. 554–563 (1992)
40. Delgado, A., Calegari, D., García, F., Weber, B.: Model-driven management of BPMN-based business process families. *Softw. Syst. Model.* **21**(6), 2517–2553 (2022)
41. Ayora, C., Torres, V., Weber, B., Reichert, M., Pelechano, V.: VIVACE: a framework for the systematic evaluation of variability support in process-aware information systems. *Inf. Softw. Technol.* **57**, 248–276 (2015)
42. La Rosa, M., Dumas, M., ter Hofstede, A.H.M., Mendling, J.: Configurable multi-perspective business process models. *Inf. Syst.* **36**, 313–340 (2011)
43. Reichert, M., Weber, B.: *Enabling Flexibility in Process-Aware Information Systems*. Springer, Berlin Heidelberg (2012)
44. Becker, J., Delfmann, P., Knackstedt, R.: Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models. In: Becker, J., Delfmann, P. (eds.) *Reference Modeling*, pp. 27–58. Physica-Verlag HD, Heidelberg (2007)
45. Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H.: Configurable Process Models—A Foundational Approach. In: Becker, J., Delfmann, P. (eds.) *Reference Modeling*, pp. 59–77. Physica-Verlag HD, Heidelberg (2007)
46. Furrer, F.J.: *Safety and Security of Cyber-Physical Systems: Engineering dependable Software using Principle-based Development*. Springer Fachmedien Wiesbaden, Wiesbaden (2022)
47. Iqintin: Requirements—IOT-A: Internet of Things Architecture. https://www.iiot-a.eu/public/requirements/copy_of_requirements (2022). Accessed 30 Nov 2022
48. Graja, I., Kallel, S., Guermouche, N., Kacem, A.H.: BPMN4CPS: A BPMN Extension for Modeling Cyber-Physical Systems. In: *2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 152–157. IEEE, Paris, France (2016)
49. Seiger, R., Huber, S., Schlegel, T.: PROtEUS: An Integrated System for Process Execution in Cyber-Physical Systems. In: Gaaloul, K., Schmidt, R., Nurcan, S., Guerreiro, S., Ma, Q. (eds.) *Enterprise, Business-Process and Information Systems Modeling*, pp. 265–280. Springer International Publishing, Cham (2015)
50. Krogstie, J.: Perspectives to Process Modeling – A Historical Overview. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) *Enterprise, Business-Process and Information Systems Modeling*, pp. 315–330. Springer, Berlin Heidelberg (2012)
51. Cabanillas, C., Knuplesch, D., Resinas, M., Reichert, M., Mendling, J., Ruiz-Cortés, A.: RALph: A Graphical Notation for Resource Assignments in Business Processes. In: Zdravkovic, J.,

- Kirikova, M., Johannesson, P. (eds.) *Advanced Information Systems Engineering*, pp. 53–68. Springer International Publishing, Cham (2015)
52. Braun, R., Esswein, W.: Towards Multi-perspective Modeling with BPMN. In: Aveiro, D., Pergl, R., Valenta, M. (eds.) *Advances in Enterprise Engineering IX*, pp. 67–81. Springer International Publishing, Cham (2015)
53. Santra, D., Choudhury, S.: C-BPMN: A Context Aware BPMN for Modeling Complex Business Process. CoRR. abs/1806.01333 (2018)
54. Grambow, G., Hieber, D., Oberhauser, R., Pogolski, C.: A Context and Augmented Reality BPMN and BPMS Extension for Industrial Internet of Things Processes. In: Marrella, A., Weber, B. (eds.) *Business Process Management Workshops*, pp. 379–390. Springer International Publishing, Cham (2022)
55. Wolter, C., Schaad, A.: Modeling of Task-Based Authorization Constraints in BPMN. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *Business Process Management*, pp. 64–79. Springer, Berlin Heidelberg (2007)
56. Terenciani, M., Paiva, D., Landre, G., Cagnin, M.I.: BPMN*—A Notation for Representation of Variability in Business Process Towards Supporting Business Process Line Modeling. Presented at the 27th International Conference on Software Engineering and Knowledge Engineering July 1 (2015)
57. Delgado, A., Calegari, D.: BPMN 2.0 based modeling and customization of variants in business process families. In: 2017 XLIII Latin American Computer Conference (CLEI), pp. 1–9. IEEE, Cordoba (2017)
58. Döhning, M., Zimmermann, B.: vBPMN: Event-Aware Workflow Variants by Weaving BPMN2 and Business Rules. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) *Enterprise, Business-Process and Information Systems Modeling*, pp. 332–341. Springer, Berlin Heidelberg (2011)
59. Park, J., Yeom, K.: A Modeling Approach for Business Processes Based on Variability. In: 2011 Ninth International Conference on Software Engineering Research, Management and Applications, pp. 211–218. IEEE, Baltimore (2011)
60. Zhang, H., Han, W., Ouyang, C.: Extending BPMN for Configurable Process Modeling 14
61. Pillat, R.M., Oliveira, T.C., Alencar, P.S.C., Cowan, D.D.: BPMNt: a BPMN extension for specifying software process tailoring. *Inf. Softw. Technol.* **57**, 95–115 (2015)
62. Chen, Y.-T., Wang, M.-S.: A study of extending BPMN to integrate IoT applications. In: 2017 International Conference on Applied System Innovation (ICASI), pp. 1797–1800. IEEE, Sapporo (2017)
63. Chang, C., Srirama, S.N., Buyya, R.: Mobile cloud business process management system for the internet of things: a survey. *ACM Comput. Surv.* **49**, 1–42 (2017)
64. Yousfi, A., Bauer, C., Saidi, R., Dey, A.K.: uBPMN: a BPMN extension for modeling ubiquitous business processes. *Inf. Softw. Technol.* **74**, 55–68 (2016)
65. Kirikkayis, Y., Gallik, F., Winter, M., Reichert, M.: BPMNE4IoT: a framework for modeling, executing and monitoring IoT-driven processes. *Future Internet*. **15**, 90 (2023)
66. Kirikkayis, Y., Gallik, F., Seiger, R., Reichert, M.: Integrating IoT-Driven Events into Business Processes. In: Cabanillas, C., Pérez, F. (eds.) *Intelligent Information Systems*, pp. 86–94. Springer International Publishing, Cham (2023)
67. Fedeli, A., Fornari, F., Polini, A., Re, B., Torres, V., Valderas, P.: FloBP: a model-driven approach for developing and executing IoT-enhanced business processes. *Softw. Syst. Model.* (2024)
68. Meyer, S., Ruppen, A., Hilty, L.: The Things of the Internet of Things in BPMN. In: Persson, A., Stirna, J. (eds.) *Advanced Information Systems Engineering Workshops*, pp. 285–297. Springer International Publishing, Cham (2015)
69. Meyer, S., Ruppen, A., Magerkurth, C.: Internet of Things-Aware Process Modeling: Integrating IoT Devices as Business Process Resources. In: King, R. (ed.) *Active Flow and Combustion Control 2018*, pp. 84–98. Springer International Publishing, Cham (2013)
70. Valderas, P., Torres, V., Serral, E.: Towards an interdisciplinary development of IoT-enhanced business processes. *Bus. Inf. Syst. Eng.* **65**, 25–48 (2023)
71. Valderas, P., Torres, V., Serral, E.: Modelling and executing IoT-enhanced business processes through BPMN and microservices. *J. Syst. Softw.* **184**, 111139 (2022)
72. Venkatakumar, H., Schmidt, W.: Subject-oriented specification of IoT scenarios. In: *Proceedings of the 11th International Conference on Subject-Oriented Business Process Management*, pp. 1–10. ACM, Seville (2019)
73. Kannengiesser, U.: Agents Implementing Subject Behaviour: A Manufacturing Scenario. In: Fleischmann, A., Schmidt, W., Stary, C. (eds.) *S-BPM in the Wild*, pp. 201–216. Springer International Publishing, Cham (2015)
74. Zerbato, F., Seiger, R., Federico, G.D., Burattin, A., Weber, B.: Granularity in Process Mining: Can we fix it?
75. Baumgräß, A., Botezatu, M., Di Ciccio, C., Dijkman, R., Grefen, P., Hewelt, M., Mendling, J., Meyer, A., Pourmirza, S., Völzer, H.: Towards a Methodology for the Engineering of Event-Driven Process Applications. In: Reichert, M., Reijers, H.A. (eds.) *Business Process Management Workshops*, pp. 501–514. Springer International Publishing, Cham (2016)
76. Schöning, S., Ackermann, L., Jablonski, S., Ermer, A.: IoT meets BPM: a bidirectional communication architecture for IoT-aware process execution. *Softw. Syst. Model.* **19**, 1443–1459 (2020)
77. Soffer, P., Hinze, A., Koschmider, A., Ziekow, H., Di Ciccio, C., Koldehofe, B., Kopp, O., Jacobsen, A., Stürmeli, J., Song, W.: From event streams to process models and back: challenges and opportunities. *Inf. Syst.* **81**, 181–200 (2019)
78. Recker, J., Rosemann, M., Krogstie, J.: Ontology- versus pattern-based evaluation of process modeling languages: a comparison. *Commun. Assoc. Inf. Syst.* **20**, 48 (2007)
79. Braun, R., Schlieter, H., Burwitz, M., Esswein, W.: Extending a Business Process Modeling Language for Domain-Specific Adaptation in Healthcare. In: *Wirtschaftsinformatik Proceedings*, pp. 468–481 (2015)
80. Braun, R., Esswein, W.: Classification of Domain-Specific BPMN Extensions. In: Frank, U., Loucopoulos, P., Pastor, Ó., Petrounias, I. (eds.) *The Practice of Enterprise Modeling*, pp. 42–57. Springer, Berlin Heidelberg (2014)
81. Zarour, K., Benmerzoug, D., Guermouche, N., Drira, K.: A systematic literature review on BPMN extensions. *Bus. Process Manag. J.* (2019)
82. Stroppi, L.J.R., Chiotti, O., Villarreal, P.D.: Extending BPMN 2.0: Method and Tool Support. In: Dijkman, R., Hofstetter, J., Koehler, J. (eds.) *Business Process Model and Notation*, pp. 59–73. Springer, Berlin Heidelberg (2011)
83. van der Aalst, W., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow patterns. *Distrib. Parallel Databases* **14**, 5–51 (2003)
84. Moody, D.: The “Physics” of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans. Softw. Eng.* **35**, 756–779 (2009)
85. Duarte, E.B., Duarte, R.B., Da Silveira, D.S.: A Semiotic Analysis of the Representativeness of BPMN Graphic Elements. In: Shishkov, B. (ed.) *Business Modeling and Software Design*, pp. 225–234. Springer Nature Switzerland, Cham (2023)
86. Riehle, D., Gross, T.: Role model based framework design and integration. In: *Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pp. 117–133. ACM, Vancouver British Columbia Canada (1998)

87. Mayr, H.C., Thalheim, B.: The triptych of conceptual modeling: a framework for a better understanding of conceptual modeling. *Softw. Syst. Model.* **20**, 7–24 (2021)
88. Kummer, T.-F., Recker, J., Mendling, J.: Enhancing understandability of process models through cultural-dependent color adjustments. *Decis. Support. Syst.* **87**, 1–12 (2016)
89. Aagesen, G., Krogstie, J.: BPMN 2.0 for Modeling Business Processes. In: vom Brocke, J. and Rosemann, M. (eds.) *Handbook on Business Process Management 1*, pp. 219–250. Springer, Berlin Heidelberg (2015)
90. Wand, Y., Weber, R.: On the deep structure of information systems. *Inf. Syst. J.* **5**, 203–223 (1995)
91. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M.: *Workflow Patterns: The Definitive Guide*. The MIT Press, Cambridge (2016)
92. *Workflow Patterns Home Page*. <http://www.workflowpatterns.com/> (2021). Accessed 12 Feb 2021
93. Penicina, L.: Linking BPMN, ArchiMate, and BWW: Perfect Match for Complete and Lawful Business Process Models? Short Pap. Proc. 6th IFIP WG 81 Work. Conf. Pract. Enterp. Model. PoEM 2013. 1023, pp. 156–165 (2013)
94. Recker, J., Indulska, M., Rosemann, M., Green, P.: The ontological deficiencies of process modeling in practice. *Eur. J. Inf. Syst.* **19**, 501–525 (2010)
95. Genon, N., Heymans, P., Amyot, D.: Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. In: Malloy, B., Staab, S., Van Den Brand, M. (eds.) *Software Language Engineering*, pp. 377–396. Springer, Berlin Heidelberg (2011)
96. Skouti, T.: Semantically consistent transformation of behavioral and structural models. In: 2023 IEEE 25th Conference on Business Informatics (CBI), pp. 1–10 (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Tarek Skouti received his Master's degree in business information systems from Technische Universität Dresden, Germany, in 2019. He is currently pursuing his PhD in computer science on the topic of a role-based unification of structural and behavioral modeling. He was a research assistant at Chair of Business Information Systems, especially Information Systems in Industry and Trade, at TU Dresden, Germany, from 2019 to 2023. His research interests are role-based modeling,

business process modeling, and trustworthy artificial intelligence.



Ronny Seiger received his Diploma in computer science from Technische Universität Dresden, Germany, in 2011, and his PhD in 2018. He was a research assistant at the Software Engineering of Ubiquitous Systems group from 2012 to 2015 and at the Software Technology group from 2015 to 2019, both at TU Dresden. From 2019 to 2022, Ronny Seiger worked as a Post-Doc at the Chair for Software Systems Programming and Development at the University

of St. Gallen, Switzerland. Since 2022, he has been an Assistant Professor for Computer Science with focus on Software Engineering Methods and Techniques at the University of St. Gallen. His research interests include software engineering and software architecture, workflows and processes, cyber-physical systems (CPS) and Internet of Things (IoT), as well as robotics and distributed systems. His main research is at the intersection of business process management (BPM), software engineering, and IoT with a focus on applying BPM technologies for analyzing and automating processes in cyber-physical systems.



Frank J. Furrer received his Master's Degree in Electrical Engineering from the Swiss Institute of Technology (Zurich, Switzerland) in 1970. He earned his PhD (Doktor der technischen Wissenschaften) from the same institution in 1975. After five years as a project leader in a software company, he founded with a partner their own company building industrial control systems and then followed 13 years as a senior consultant for a global financial institution's worldwide

information system architecture. After his 'retirement' in 2013, he was invited to teach software architecture at the computer science faculty of the Technical University of Dresden. He was awarded the title of honorary professor in 2015 and emerited in 2024. Prof. Dr. F.J. Furrer is the author of 6 textbooks and numerous technical papers.



Susanne Strahringer is a professor of Business Information Systems, especially Information Systems in Industry and Trade, at TU Dresden, Germany. Before joining TU Dresden, she held positions at the University of Augsburg and the European Business School. She graduated from the Darmstadt University of Technology, where she also obtained her PhD and completed her habilitation thesis. She has published in Information and Management, Information Sys-

tems Management, The Data Base for Advances in Information Systems, and Business and Information Systems Engineering, among others. Her research interests focus on IS management, ERP systems, and enterprise modeling.